# MULTI-LABEL AND SEMI-SUPERVISED IMAGE CLASSIFICATION

Marjan Stoimchev

**Doctoral Dissertation**
**Jožef Stefan International Postgraduate School**
**Ljubljana, Slovenia**

**Supervisor:** Prof. Dr. Sašo Džeroski, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
**Co-Supervisor:** Dr. Dragi Kocev, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
**Co-Supervisor:** Dr. Jurica Levatić, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia

**Evaluation Board:**
Asst. Prof. Panče Panov, Chair, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Krištof Oštir, Member, University of Ljubljana, Faculty of Civil and Geodetic Engineering, Ljubljana, Slovenia
Dr. Dino Ienco, Member, French National Research Institute for Agriculture, Food and Environment (INRAE), UMR TETIS, Montpellier, France

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL




Marjan Stoimchev

# MULTI-LABEL AND SEMI-SUPERVISED IMAGE CLASSIFICATION

**Doctoral Dissertation**

# VEČOZNAČNA IN DELNO NADZOROVANA KLASIFIKACIJA SLIK

**Doktorska disertacija**

**Supervisor:** Prof. Dr. Sašo Džeroski

**Co-Supervisor:** Dr. Dragi Kocev

**Co-Supervisor:** Dr. Jurica Levatić

Ljubljana, Slovenia, December 2025

*To Viktorija and my parents*

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisors for their invaluable guidance throughout this PhD journey. I am profoundly thankful to Prof. Dr. Sašo Džeroski for his outstanding mentorship, vision, and encouragement over the years. His trust, constructive feedback, and unwavering support have been instrumental in shaping my research and professional development. I am deeply grateful for the opportunity to work under his supervision and to learn from his vast expertise, scientific rigor, and dedication to advancing the field of artificial intelligence.

I extend my sincere appreciation to Dr. Dragi Kocev for his insightful guidance and critical perspective, which have significantly contributed to the depth and quality of my work. His expertise, thoughtful feedback, and encouragement have played a central role in refining my ideas and research direction.

I am especially thankful to Dr. Jurica Levatić for his exceptional guidance, patience, and constant support throughout my studies. His willingness to engage in in-depth discussions and to provide timely advice has been invaluable. Our many conversations have greatly enriched my understanding and shaped both the technical and conceptual aspects of my research.

I would also like to thank my colleagues at the Department of Knowledge Technologies for their support and collaboration, creating an environment that fostered both learning and growth throughout my PhD studies.

Most importantly, my deepest gratitude goes to Viktorija, my partner, who has been my pillar of strength throughout this entire journey. She taught me resilience and patience, provided me with valuable critiques during the most challenging times, and stood by me when I needed support the most. Without her unwavering presence and encouragement during the hardest moments of my PhD studies, this achievement would not have been possible.

I am profoundly grateful to my mother, Veselinka, for her unconditional love, selflessness, and unwavering belief in me. Her strength, perseverance, and endless sacrifices have made everything I have achieved possible. She has been my greatest source of inspiration, a constant reminder of what dedication, kindness, and resilience truly mean. Her encouragement during moments of doubt, her wisdom in times of uncertainty, and her boundless care have guided me through every step of this journey. No words can fully express how deeply I appreciate all that she has done for me and the example she has set through her courage and compassion.

Finally, I wish to express my deepest love and gratitude to my grandmother, Dušanka, whose wisdom, warmth, and boundless care have been a guiding light in my life. Her faith in me and her lifelong example of resilience and compassion continue to inspire me every day.

To all those who have been part of this journey, though not mentioned individually, your support and kindness have been deeply appreciated and will always be remembered.

# Abstract

The rapid expansion of Earth observation (EO) systems has generated unprecedented volumes of satellite imagery, creating both remarkable opportunities and significant challenges for environmental monitoring and land cover analysis. While this abundance of data enables comprehensive global monitoring, extracting actionable insights requires sophisticated analytical methods capable of modeling the inherent complexity of Earth's landscapes, where multiple land cover types and complex semantic relationships often coexist within a single image. Traditional supervised single-label classification approaches prove inadequate for such scenarios, necessitating structured output prediction frameworks that can capture multi-label dependencies and taxonomic constraints while operating effectively under limited supervision.

This dissertation addresses the critical intersection of structured output prediction and varying levels of supervision in remote sensing image analysis. The research focuses on developing novel semi-supervised learning methodologies for multi-label classification (MLC) and hierarchical multi-label classification (HMLC) that effectively leverage abundant unlabeled satellite imagery alongside limited labeled samples while preserving complex label dependencies inherent in real-world visual content.

We develop a hybrid semi-supervised framework that combines convolutional neural network-based feature extraction with semi-supervised predictive clustering trees (SSL-PCTs) and random forests (SSL-RForest). This two-step approach leverages the representational power of deep learning while maintaining the flexibility of tree-based methods for structured output prediction. Through extensive evaluation across multiple remote sensing datasets, we demonstrate that SSL-PCT and SSL-RForest consistently outperform their supervised counterparts across different network architectures and proportions of labeled data, with SSL-RForest emerging as particularly effective compared to state-of-the-art self-supervised and semi-supervised methods. To enhance the computational efficiency of this framework, we investigate dimensionality reduction techniques, achieving dramatic speed-ups of 2-3 orders of magnitude in training times while maintaining or improving predictive performance, making the approach more practical for large-scale applications.

Motivated by the computational inefficiencies and suboptimal feature alignment inherent in two-stage learning approaches, we introduce SSL-MAE: a semi-supervised learning framework for multi-label classification of remote sensing images based on masked autoencoders. This unified end-to-end framework overcomes the limitations of traditional separate pre-training and fine-tuning phases by integrating self-supervised representation learning with task-specific discriminative learning through adaptive joint optimization. Using masked autoencoders as the self-supervised component, this framework dynamically balances supervision signals throughout training, achieving competitive performance against two-stage methods while significantly improving computational efficiency. The adaptive joint learning mechanism demonstrates broad applicability, successfully enhancing other state-of-the-art self-supervised techniques, including Bootstrap Your Own Latent (BYOL) and Momentum Contrast (MoCo).

Recognizing that real-world land cover taxonomies often exhibit complex multi-path hierarchical relationships that existing methods fail to capture adequately, we present ierarchical and Explicit Label Modeling (HELM) – the first semi-supervised image-based method capable of modeling multi-path hierarchies in hierarchical multi-label classification. The framework integrates hierarchy-specific class tokens with graph-based structure encoders to capture complex relationships among labels, incorporating BYOL as a self-supervised learning (Self-SL) module to effectively leverage unlabeled data while maintaining hierarchical consistency. Through evaluation across remote sensing and medical imaging domains, we demonstrate that incorporating hierarchical information significantly improves predictive performance in both supervised and semi-supervised settings.

Through comprehensive evaluation across multiple datasets, we demonstrate that the proposed methods consistently outperform state-of-the-art supervised and semi-supervised approaches while requiring significantly less labeled data. The developed frameworks address fundamental challenges in leveraging unlabeled data for structured prediction tasks in remote sensing domain, offering practical solutions for large-scale monitoring applications where annotation costs are prohibitive. The work establishes new benchmarks and resources for the research community while opening promising directions for future investigation in automatic hierarchy discovery, visual-semantic alignment, and multi-modal learning scenarios.

# Povzetek

Hitra širitev sistemov za opazovanje Zemlje je povzročila nastanek neprimerljivo velike količine satelitskih posnetkov, kar prinaša izjemne priložnosti in hkrati pomembne izzive za spremljanje okolja ter analizo pokrovnosti tal. Čeprav ta obilica podatkov omogoča celovito globalno spremljanje, njihova pretvorba v uporabne informacije zahteva napredne analitične metode, sposobne modelirati kompleksnost zemeljskih pokrajin, kjer se različni tipi pokrovnosti tal pogosto pojavljajo hkrati v posameznih odsekih posnetkov in tvorijo hierarhične semantične odnose. Tradicionalni pristopi enoznačne klasifikacije se v takih primerih izkažejo za neustrezne, zato so potrebni modeli napovedovanja strukturiranih izhodov, ki omogočajo zajem odvisnosti med oznakami in upoštevanje taksonomskih omejitev ter hkrati učinkovito delovanje pri omejeni količini označenih podatkov.

Disertacija naslavlja presečišče napovedovanja strukturiranih izhodov in različnih stopenj nadzora pri analizi posnetkov daljinskega zaznavanja. Raziskava se osredotoča na razvoj novih polnadzorovanih učnih metodologij za večoznačno klasifikacijo in hierarhično večoznačno klasifikacijo, ki izkoriščajo obilico neoznačenih satelitskih posnetkov skupaj z omejenim številom označenih primerov ter hkrati ohranjajo kompleksne odvisnosti med oznakami, značilne za vizualno vsebino iz resničnega sveta.

Razvit je hibridni polnadzorovan okvir, ki združuje ekstrakcijo značilk na osnovi konvolucijskih nevronskih mrež s polnadzorovanimi napovednimi gručevalnimi drevesi in ansambelskimi metodami. Ta dvostopenjski pristop izkorišča predstavitveno moč globokega učenja in hkrati ohranja prilagodljivost metod na osnovi dreves za napovedovanje strukturiranih izhodov. Obsežna evalvacija na več podatkovnih zbirkah daljinskega zaznavanja pokaže, da SSL-PCT in SSL-RForest dosledno presegata primerljive nadzorovane pristope pri različnih mrežnih arhitekturah in deležih označenih podatkov, pri čemer se SSL-RForest izkaže za posebej učinkovitega v primerjavi z najsodobnejšimi samonadzorovanimi in polnadzorovanimi metodami. Za izboljšanje računske učinkovitosti tega okvira so preučene tehnike redukcije dimenzionalnosti, ki omogočajo pohitritve učenja za 2–3 velikostne rede ob ohranjeni ali celo izboljšani napovedni zmogljivosti, kar pristop naredi primeren za obsežno operativno uporabo.

Zaradi računskih neučinkovitosti in nepopolnega usklajevanja značilk, značilnih za dvostopenjske pristope, je predstavljen SSL-MAE, enoten end-to-end okvir, ki odpravlja omejitve tradicionalnega ločenega predusposabljanja in fine-tuninga z integracijo samonadzorovanega predstavitvenega učenja z diskriminativnim učenjem, specifičnim za nalogo, preko adaptivne skupne optimizacije. Z uporabo maskiranih samokodirnikov kot samonadzorovane komponente okvir dinamično uravnava nadzorne signale skozi celoten proces učenja ter dosega konkurenčno zmogljivost v primerjavi z dvostopenjskimi metodami, ob hkratnem bistvenem izboljšanju računske učinkovitosti. Adaptivni mehanizem skupnega učenja se izkaže za široko uporabnega in uspešno izboljša tudi druge samonadzorovane pristope, kot sta BYOL in MoCo.

Ker realne taksonomije pokrovnosti tal pogosto vsebujejo kompleksne večpotne hierarhične odnose, ki jih obstoječe metode ne obravnavajo ustrezno, je predstavljen HELM, prvi

polnadzorovan slikovni pristop, sposoben modelirati večpotne hierarhije v hierarhični večoznačni klasifikaciji. Okvir združuje hierarhično-specifične razredne žetone z grafovskimi strukturnimi kodirniki za zajem kompleksnih odnosov med oznakami ter vključuje BYOL kot samonadzorovan učni modul za učinkovito izrabo neoznačenih podatkov ob ohranjanju hierarhične skladnosti. Evalvacija na področjih daljinskega zaznavanja in medicinskega slikanja pokaže, da vključevanje hierarhičnih informacij pomembno izboljša napovedne zmogljivosti tako v nadzorovanih kot tudi polnadzorovanih nastavitvah.

Celovita evalvacija na več podatkovnih zbirkah potrjuje, da predlagane metode dosledno presegajo najsodobnejše nadzorovane in polnadzorovane pristope ter zahtevajo bistveno manj označenih podatkov. Predstavljeni okviri naslavljajo ključne izzive izkoriščanja neoznačenih podatkov pri nalogah napovedovanja strukturiranih izhodov in ponujajo praktične rešitve za obsežno spremljanje okolja, kjer so stroški označevanja visoki. Delo postavlja nove primerjalne standarde in vire za raziskovalno skupnost ter odpira obetavne smeri prihodnjih raziskav na področjih samodejnega odkrivanja hierarhij, vizualno-semantičnega usklajevanja in večmodalnega učenja.

# Contents

**List of Figures** **xv**

**List of Algorithms** **xvii**

**Abbreviations** **xix**

**1 Introduction** **1**
1.1 Motivation . . . 3
1.2 Purpose of the Disseration . . . 4
1.3 Goals and Hypotheses . . . 4
1.4 Methodology . . . 6
1.5 Contributions . . . 7
1.6 Organization of the Thesis . . . 8

**2 Background and Related Work** **9**
2.1 SOP in Image Classification . . . 9
2.1.1 The task of multi-label image classification . . . 9
2.1.2 The task of hierarchical multi-label image classification . . . 10
2.2 Deep Learning for SOP . . . 11
2.2.1 Feature extractors . . . 11
2.2.1.1 Convolutional neural networks . . . 11
2.2.1.2 Vision transformers . . . 13
2.2.2 Graph neural networks . . . 14
2.3 Learning with Varying Levels of Supervision . . . 16
2.3.1 Supervised learning . . . 17
2.3.2 Semi-supervised learning . . . 18
2.3.2.1 Consistency regularization approaches . . . 20
2.3.2.2 Pseudo-labeling and self-training approaches . . . 21
2.3.2.3 Hybrid approaches . . . 21
2.3.2.4 PCTs for SSL . . . 22
2.3.3 Self-supervised learning . . . 23
2.3.3.1 Contrastive learning approaches . . . 25
2.3.3.2 Generative reconstruction approaches . . . 26
2.4 HMLC of RSI with Semi-Supervised Learning . . . 28

**3 Deep Network Architectures as Feature Extractors for MLC of RSIs** **31**
Paper: Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images . . . 33

**4 Semi-Supervised MLC of RSIs With PCTs and Ensembles** **57**
4.1 Semi-Supervised MLC of RSIs With PCTs and Ensembles . . . 57

Paper: Semi-Supervised Multi-Label Classification of Land Use/Land Cover in Remote Sensing Images With Predictive Clustering Trees and Ensembles . . 59
4.2 Dimensionality Reduction for Efficient SSL . . . . . 75
Paper: Dimensionality Reduction for Efficient Semi-Supervised Learning From Remote Sensing Images . . . . . 76

**5 SSL-MAE: Adaptive SSL for Multi-Label RSI Classification 87**
Paper: SSL-MAE: Adaptive Semi-Supervised Learning Framework for Multi-Label Classification of Remote Sensing Images Using Masked Autoencoders 89

**6 HMLC of RSI with Supervised and Semi-Supervised Learning 105**
6.1 HMLC of RSI . . . . . 105
Paper: Hierarchical Multi-label Classification of Land Use/Land Cover in Remote Sensing Images . . . . . 107
6.2 Hierarchical Multi-Label Image Classification with Graph Learning . . . . . 117
Paper: Fully- and Semi-Supervised Hierarchical Multi-Label Image Classification with Graph Learning . . . . . 118

**7 Conclusions and Future Work 145**
7.1 Summary of Contributions . . . . . 145
7.2 Further Work . . . . . 147

**References 151**

**Bibliography 157**

**Biography 159**

# List of Figures

2.1 Evolution of image classification paradigms using a remote sensing image as example. Left: Multi-class classification assigns a single dominant label. Center: MLC predicts multiple co-occurring labels. Right: HMLC arranges outputs within a taxonomy, where predicting a leaf label entails predicting all its ancestors, ensuring logical consistency in the label space. . . . . 10

2.2 A simple CNN architecture. The network progressively transforms the input image through alternating convolution and pooling layers, extracting increasingly abstract features. The final dense layers aggregate these hierarchical features into a compact latent representation. . . . . 12

2.3 Vision Transformer architecture. An input image is divided into fixed-size patches, which are linearly embedded and combined with positional encodings. These patch embeddings are processed by a transformer encoder consisting of alternating multi-head attention and MLP blocks. The architecture enables global interactions between all patches from the first layer, with the final representation used for classification tasks. 14

2.4 Graph Neural Network pipeline for hierarchical multi-label classification. The underlying label hierarchy $\mathcal{H}$ (left) is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes correspond to labels and edges encode parent–child relationships. Initial node embeddings derived from the visual encoder $f_\theta$ are iteratively refined through multiple graph convolutional layers, producing hierarchy-aware embeddings that respect structural dependencies. This structured embedding process ensures semantically and hierarchically consistent multi-label predictions. . . . . 16

2.5 Comparison of different learning paradigms: (a) Supervised learning uses only labeled data, (b) Semi-supervised learning combines labeled and unlabeled data, (c) Self-supervised learning uses pretext tasks for knowledge transfer to target tasks. . . . . 17

2.6 Core SSL paradigms. (a) Consistency regularization enforces prediction stability under different perturbations or augmentations of the same input. (b) Pseudo-labeling assigns high-confidence model predictions as provisional labels for unlabeled data, enabling iterative self-training. 19

2.7 Contrastive learning framework for visual representation learning. The architecture processes pairs of images through shared visual encoders $f_\theta$ followed by projection heads (MLP) to generate normalized embeddings in a hyperspherical space. For each anchor image $x_i$, augmented views $t$ and $t'$ are created and mapped to nearby points in the embedding space. Simultaneously, embeddings from different images are pushed apart. This contrastive objective learns discriminative visual representations by maximizing agreement between semantically similar views while minimizing similarity to negative samples, resulting in a structured embedding space where semantically related images cluster together. . . . . 26

2.8 Masked Autoencoder (MAE) self-supervised pre-training framework. The input image is divided into patches and a large subset (typically 75%) is randomly masked (shown in gray). The encoder $f_\theta$ processes only the visible patches (shown in green/dark patches), creating a compact representation. The decoder $g_\theta$ then reconstructs the original image from this partial encoding, learning to predict the masked patches (shown in pink/red in the target). This asymmetric encoder-decoder architecture enables efficient pre-training by processing only a fraction of the input while learning rich visual representations through the challenging reconstruction task of predicting missing visual content. . . . . 28

# List of Algorithms

2.1 SimCLR's main learning algorithm. . . . . 27
2.2 Masked Autoencoder (MAE) Training Algorithm. . . . . 29

# Abbreviations

AI ... Aritifial Intelligence
RS ... Remote Sensing
RSI ... Remote Sensing Image
SOP ... Structured Output Prediction
DAG ... Direct Acyclic Graph
MCC ... Multi-Class Classification
MLC ... Multi-Label Classification
HMLC.. Hierarchical Multi-Label Classification
SL ... Supervised Learning
SSL ... Semi-Supervised Learning
Self-SL.. Self-Supervised Learning
PL ... Pseudo Labeling
EO ... Eearth Observation
CNN ... Convolutional Neural Network
GNN ... Graph Neural Netowrk
GCN ... Graph Convolutional Network
GAT ... Graph Attention Network
ViT ... Vision Transformer
PCTs ... Predictive Clustering Trees
BYOL .. Bootstrap Your Own Latent
MoCo... Momentum Contrast
MAE ... Masked Autoencoder
MIM ... Masked Image Modeling
CE ... Cross Entropy
BCE ... Binary Cross Entropy

# Chapter 1

# Introduction

Artificial intelligence (AI) and deep learning have revolutionized the way we handle and analyze large-scale data, particularly in remote sensing (RS), where imagery is produced at an ever-accelerating pace. For instance, the European Space Agency's Copernicus Sentinel program alone generates tens of terabytes of data daily, reaching a cumulative download volume of 586 PiB (pebibytes) by October 2023[1]. RS provides a powerful means of studying the Earth's surface and its various phenomena from a distance, typically via satellite or aerial imagery, eliminating the need for direct physical contact. This capability enables large-scale environmental monitoring, disaster assessment, and land cover analysis.

Land cover mapping stands out as one of the most widely explored areas of RS, supported by large public datasets that provide images and corresponding labels [1]. Accurate land cover maps are critical for tracking climate change, monitoring urban growth and pollution, and assessing the impacts of extreme weather events, ocean warming, ice loss, and $CO_2$ emissions. They also inform sustainable city planning, infrastructure development, and agricultural practices through optimized water use and soil management. By providing near-real-time insights, RS not only supports informed decision-making for environmental protection and disaster response but also promotes more efficient and sustainable resource management [2].

Despite the remarkable progress of AI-based approaches in remote sensing image (RSI) analysis, several gaps remain. Traditional single-label classification methods, which map each image to a single category, have shown strong performance [3], [4], but they are insufficient in real-world contexts where multiple land cover types frequently coexist. This has motivated a shift toward *multi-label classification* (MLC), which assigns multiple relevant categories to each image [5], and further toward *hierarchical multi-label classification* (HMLC), which incorporates taxonomic relationships among multiple labels [6], [7]. For instance, standardized systems such as CORINE Land Cover [8] define structured label spaces where fine-grained classes (e.g., *coniferous forest*) are nested under broader ones (e.g., *forest*, *natural vegetation*). These structures improve interpretability and semantic consistency but also introduce additional modeling challenges.

A major bottleneck arises from the scarcity of annotated data, which is even more pronounced in MLC and HMLC tasks: annotators must identify multiple coexisting land cover types per image and ensure consistency across hierarchical levels, making large-scale dataset creation prohibitively costly and time-consuming. With the increasing demand for accurate, large-scale RSI analysis, computer vision and deep learning methods are being explored to enhance MLC and HMLC by capturing spatial and semantic relationships in

[1]Sentinel Data Access Annual Report 2023, ESA. Available at: `https://sentiwiki.copernicus.eu/__attachments/1673407/COPE-SRCO-RP-2400521%20-%20Sentinel%20Data%20Access%20Annual%20Report%202023%20-%201.1.pdf`

images. Yet their effectiveness remains constrained by the availability of labeled data.

This limitation calls for learning paradigms that reduce reliance on labeled data while effectively leveraging the abundance of unlabeled imagery. Semi-supervised learning (SSL) has emerged as a promising approach by combining a small set of labeled samples with large quantities of unlabeled data [9], [10]. SSL techniques such as consistency regularization [11], pseudo-labeling [12], and self-training [13] have demonstrated effectiveness in single-label classification, but extending them to MLC and HMLC is non-trivial due to complex label dependencies and hierarchical constraints. Complementarily, self-supervised learning (Self-SL) eliminates the need for labels entirely during representation learning by relying on pretext tasks such as contrastive learning [14], [15] or masked image modeling [16], offering transferable features that can significantly improve downstream RS applications. The interplay between SSL and Self-SL is increasingly recognized as synergistic: self-supervised pretraining provides strong initialization, while SSL focuses learned representations toward task-specific objectives when limited labels are available.

In this thesis, we address these challenges by focusing on two orthogonal dimensions that are critical for advancing RSI analysis. The first dimension concerns the structuring of the output space, where we move beyond the limitations of single-label classification by exploring MLC and HMLC frameworks. These approaches capture the inherent complexity of real-world remote sensing scenes, where multiple semantic classes such as land cover types, land use patterns, and man-made or natural objects coexist and exhibit hierarchical relationships. The second dimension addresses varying levels of supervision, recognizing that the scarcity of annotated data remains a significant bottleneck. By developing novel semi-supervised methodologies tailored for RSI, we aim to leverage the abundance of unlabeled data while maximizing the utility of limited labeled examples. Through the integration of these two dimensions – structured output prediction (SOP) and flexible supervision paradigms – this research aims to develop more robust and generalizable models that can capture complex label dependencies, adapt to limited data regimes, and enhance large-scale environmental monitoring and sustainable resource management.

The methodologies developed in this dissertation provide concrete solutions to two longstanding challenges in remote sensing: the scarcity of annotated data and the need for hierarchically consistent land cover representations. The proposed semi-supervised learning (SSL) frameworks directly address the well-recognized *label scarcity bottleneck* [17], where the cost and time associated with expert annotation make large-scale, high-resolution mapping projects prohibitively demanding [18]. Using vast amounts of unlabeled remote sensing data, the developed methods enable experts to produce accurate and reliable land cover maps with only a limited set of ground-truth samples. In parallel, the hierarchy-aware components of the proposed frameworks tackle the pervasive problem of semantic inconsistency commonly observed in land cover maps, where predictions may violate logical taxonomic constraints, such as identifying *vineyards* without the parent category *agricultural areas* [19], [20]. By embedding established nomenclatures such as the CORINE Land Cover classification directly into the learning process, the models ensure hierarchical coherence and improve robustness through adaptive fallback mechanisms to broader category levels under uncertainty. Collectively, these advancements yield reliable, semantically consistent, and multi-scale mapping tools that support large-scale environmental monitoring and geospatial decision-making, thereby addressing a critical gap in the current state of research [21], [22].

The core principles of this dissertation, particularly the utilization of unlabeled data and the enforcement of semantic structure, are inherently domain-agnostic. These principles hold substantial practical relevance in other fields that face similar data-related challenges, most notably in medical imaging. In domains such as digital pathology and

radiology, the acquisition of expert annotations from certified specialists is both costly and time-consuming, resulting in a pronounced *label scarcity* bottleneck that can be effectively mitigated through semi-supervised learning [23], [24]. Moreover, medical diagnosis tasks are inherently hierarchical, adhering to well-defined taxonomies for disease classification in modalities such as chest X-rays [25] or dermatological imaging [26], and extending to related biomedical areas such as protein function prediction [27]. The semi-supervised strategies developed in this work, which align with recent advances in hierarchical medical learning [24], enable the construction of robust models from large archives of unlabeled medical images, including magnetic resonance and histopathology data. In parallel, hierarchy-aware frameworks ensure clinically consistent predictions by embedding domain-specific taxonomies, such as the International Classification of Diseases (ICD-10) [28], directly into the learning process. Together, these advancements provide medical researchers and clinicians with data-efficient, interpretable, and reliable tools for supporting diagnostic decision-making.

## 1.1 Motivation

The motivation for this thesis is threefold.

**1. Advancing SOP in RS.** There is a pressing need to extensively explore and evaluate SOP methods in RSI classification, particularly for land cover mapping, where complex label relationships require sophisticated modeling approaches. RS imagery frequently contains multiple coexisting land cover types with hierarchical relationships organized within taxonomic structures like the CORINE nomenclature [8]. While recent advances in MLC have introduced graph neural networks (GNNs) [29], attention mechanisms, and transformer-based approaches [30], these methods have not been comprehensively evaluated in this context. Furthermore, there remains limited exploration of how classical methods, particularly predictive clustering trees (PCTs) [31], can be effectively combined with deep feature extractors to model these complex label dependencies. We aim to investigate hybrid methodologies that leverage the efficiency of traditional methods alongside the representational power of deep learning architectures.

**2. Addressing gaps in SSL for structured image classification tasks.** A second motivation stems from the fundamental gap in SSL methods for structured image classification tasks. While SSL has achieved remarkable success in conventional multi-class settings through techniques such as consistency regularization [11] and pseudo-labeling, these approaches are predominantly designed for single-label problems and do not adequately address the unique challenges of MLC and HMLC in remote sensing imagery. Although SSL has been extensively studied for classical data mining and other structured prediction tasks [9], it remains underexplored in the context of image-based SOP, where the need for exploiting unlabeled data is even greater. Global models for SOP have been shown to yield better predictive performance than local decomposition approaches, as they are more computationally efficient, produce simpler models, and tend to overfit less [32]. This limitation motivates us to explore a paradigm shift toward SSL approaches grounded in Self-SL, which offer greater flexibility by eliminating the need for labeled data during representation learning. Rather than adopting the conventional two-stage pipeline of self-supervised pre-training followed by supervised fine-tuning, we aim to develop unified solutions that integrate self-supervised representation learning with discriminative optimization for task-specific objectives, leveraging GNNs to capture hierarchical dependencies while effectively utilizing unlabeled data.

**3. Toward unified and efficient end-to-end frameworks.** The third motivation arises from the limitations of existing approaches that decouple representation learning from task-specific objectives. Such separation can hinder the effective transfer of learned features and lead to suboptimal adaptation to downstream tasks. Our goal is therefore to improve methodological efficiency and consistency by jointly optimizing representation learning and discriminative modeling within a single training process.

This unified approach offers several key advantages:

- Combining self-supervised and discriminative objectives within a single framework to promote stronger task alignment and more efficient representation learning for SOP.
- Jointly optimizing representation learning and task-specific objectives to produce features that are inherently better suited for the hierarchical and multi-label structure of SOP tasks.
- Providing a modular and extensible design that supports seamless adaptation across domains and SOP problem settings, improving both flexibility and computational efficiency.

## 1.2 Purpose of the Disseration

The purpose of this dissertation is to:

**P1** Advance the underexplored field of SSL for SOP in image analysis by designing methods that effectively leverage unlabeled data, while capturing complex label dependencies and improving generalization.

**P2** Extend PCTs and ensemble learning to handle image classification by integrating them with deep feature extraction, enhancing their effectiveness in MLC under limited supervision.

**P3** Design an end-to-end semi-supervised framework for image classification that unifies self-supervised representation learning with discriminative modeling, eliminating the need for separate pre-training and fine-tuning, and improving computational efficiency.

**P4** Advance HMLC for images by incorporating graph-based learning techniques capable of handling multi-path hierarchies and adapting to domain-specific taxonomies such as the CORINE Land Cover nomenclature.

## 1.3 Goals and Hypotheses

The overarching goal of this dissertation is to advance SSL for SOP in the context of remote sensing image analysis, addressing the pressing challenge of limited labeled data in large-scale environmental monitoring tasks. This research is motivated by the need to model complex label structures – such as co-occurring classes and hierarchical taxonomies – while leveraging the abundance of unlabeled remote sensing image data.

When designing the novel methods, we had the following specific goals in mind:

- Designing SSL methods that are specifically tailored for SOP tasks, aiming to preserve structured label dependencies while improving generalization by effectively utilizing both labeled and unlabeled data.

- Enhancing PCTs and ensemble models by integrating them with deep feature extraction techniques to improve the performance of semi-supervised MLC methods, particularly in scenarios where traditional shallow models may fall short.
- Developing a unified end-to-end SSL framework that eliminates the need for separate pretraining and fine-tuning processes by combining self-supervised representation learning with task-specific discriminative learning in a single optimization loop, improving computational efficiency while maintaining or exceeding predictive performance.
- Developing hierarchy-aware semi-supervised methods for HMLC that incorporate graph-based learning techniques to explicitly model structured label dependencies and support multi-path hierarchies, aligning with real-world taxonomies such as the CORINE Land Cover nomenclature.
- Extensively evaluating the newly proposed methods using standard benchmark datasets and comparing them to current state-of-the-art approaches across different structured output prediction tasks.
- Implementing the methods in modular frameworks that enable easy integration and scalability across different SOP tasks without requiring major architectural changes, ensuring seamless transition between MLC and HMLC scenarios.

Together, these goals form a comprehensive research agenda aimed at advancing the state-of-the-art in semi-supervised structured output learning for remote sensing imagery, with applications in land cover mapping, environmental monitoring, and sustainable resource management.

Closely related to the above goals, we have formulated and investigated the following hypotheses:

**H1** Convolutional neural network (CNN) architectures used as feature extractors combined with tree ensemble methods can achieve competitive or superior performance compared to end-to-end CNN approaches for MLC, and lightweight network architectures can achieve better performance than heavier architectures while being more computationally efficient.

**H2** Extending supervised methods for SOP to a semi-supervised setting improves model performance by effectively leveraging unlabeled data while preserving structured label dependencies in (remote sensing) image classification.

**H3** Semi-supervised methods that integrate deep feature extraction from images with predictive clustering frameworks outperform their supervised counterparts in MLC tasks.

**H4** End-to-end semi-supervised learning frameworks for image analysis that unify self-supervised representation learning with task-specific discriminative learning provide better generalization and computational efficiency than traditional two-stage SSL approaches. Moreover, incorporating diverse self-supervised pretext tasks enhances adaptability and performance.

**H5** Incorporating hierarchical information in multi-label image classification improves predictive performance by capturing structured label dependencies. Moreover, extending this with graph-based reasoning and semi-supervised learning further enhances generalization and robustness across diverse domains.

## 1.4 Methodology

To address the challenges at the intersection of SOP and varying levels of supervision, this dissertation adopts a staged methodology that integrates deep learning, classical machine learning, and Self-SL within a unified research framework. The overall approach is guided by a progression from foundational analysis to hybrid modeling, and finally to fully integrated end-to-end systems. This progression allows for systematic exploration of different architectures and learning paradigms, culminating in the development of scalable and generalizable models for semi-supervised multi-label and hierarchical MLC in RS and beyond.

The methodology is organized into four key phases:

**Phase 1: Baseline Evaluation of Deep and Classical Models for Supervised MLC.** We begin with a comprehensive evaluation of CNN architectures – including VGG [33], EfficientNet [34], and ResNet [35] – alongside classical machine learning methods for MLC in remote sensing imagery. These models will be assessed both as standalone classifiers and as feature extractors. The goal of this phase is to identify which architectures generate the most informative and transferable representations. The insights gained will guide the selection of backbone models for subsequent hybrid and end-to-end frameworks.

**Phase 2: Development of a Two-Step Semi-Supervised Framework** Based on the findings of the first phase, we will develop a hybrid semi-supervised framework that bridges deep learning and classical machine learning paradigms. This framework integrates deep CNN-based feature extraction from images with semi-supervised PCTs and their ensemble variants [36]. Pre-trained CNNs will be used to extract high-level features from remote sensing images, which will then be passed to semi-supervised PCTs to learn from both labeled and unlabeled data. This two-step architecture provides a modular and interpretable approach to semi-supervised learning while leveraging the strengths of both paradigms.

**Phase 3: End-to-End Semi-Supervised Framework for Image Classification** To address the limitations of the two-step approach, we will design a fully integrated end-to-end semi-supervised learning framework that jointly optimizes self-supervised representation learning and task-specific discriminative objectives. The core of this framework will be based on masked autoencoders [16], which provide rich, spatially aware representations of unlabeled imagery. Adaptive learning strategies will be incorporated to dynamically balance the contributions of self-supervised and supervised signals during training. Additionally, we will explore the flexibility of the framework by integrating alternative self-supervised methods such as *Bootstrap Your Own Latent* (BYOL) [15] and *Momentum Contrast* (MoCo) [37], enhancing its adaptability across different datasets and tasks.

**Phase 4: Hierarchy-Aware Semi-Supervised Learning for HMLC of Images** The final phase focuses on HMLC, where we will develop a hierarchy-aware SSL approach capable of capturing both intra-level and inter-level label dependencies. The framework will incorporate hierarchy-specific class tokens and utilize graph convolutional networks (GCNs) [38] to model taxonomic relationships, particularly those involving multi-path hierarchies. We will implement structured taxonomies based on the CORINE Land Cover nomenclature [8] for remote sensing applications and construct hierarchical label sets in medical imaging to evaluate cross-domain generalizability. A self-supervised component

will be integrated into this architecture to improve feature representations in data-scarce scenarios, thereby enhancing the model's robustness and generalization.

**Evaluation Strategy** The evaluation of all proposed methods will include both quantitative and qualitative assessments across remote sensing and medical imaging datasets. Quantitative evaluation will benchmark the models against state-of-the-art supervised and semi-supervised SOP methods, focusing on predictive performance, computational efficiency, and consistency with hierarchical label structures. Qualitative evaluation will include visualization of learned embeddings, error analysis, and interpretability of hierarchical outputs. Ablation studies will be conducted to isolate the contribution of individual components (e.g., self-supervised module, graph-based hierarchy encoder), and sensitivity analyses will be performed to understand the impact of varying proportions of labeled data. All implementations will be developed using the PyTorch ecosystem [39] and supporting libraries, ensuring scalability, reproducibility, and extensibility for future research. In line with the principles of open science, all developed resources (including modular pipelines, source code, and documentation) will be made publicly available to the research community to foster transparency, reusability, and collaboration.

## 1.5 Contributions

The scientific contributions of this dissertation focus on advancing SSL for SOP, with an emphasis on MLC and HMLC in the context of image analysis. The developed methods address key challenges in leveraging unlabeled data while preserving structured label dependencies, particularly in RSI classification. The expected contributions are as follows:

**C1** Development of a semi-supervised learning framework that integrates classical machine learning methods, specifically semi-supervised PCTs and ensemble methods, with deep learning-based feature extraction from images to form a unified two-step SSL approach. This hybrid methodology leverages the advantages of both paradigms to improve predictive performance in semi-supervised MLC of images while maintaining computational efficiency.

**C2** Design and implementation of an end-to-end SSL framework for image classification that eliminates the limitations of traditional two-step approaches by unifying self-supervised representation learning and task-specific discriminative learning. The proposed framework provides a flexible and adaptive structure that allows for seamless integration of different self-supervised pretext tasks, enhancing generalization and improving upon existing Self-SL methods.

**C3** Development of hierarchy-aware SSL methods for HMLC of images that incorporate graph-based learning techniques to explicitly model structured label dependencies and multi-path hierarchies. Unlike existing approaches that assume a single path per instance, the proposed methods enable a more accurate representation of real-world hierarchical relationships.

**C4** Adaptation of the developed methods to structured taxonomies such as the CORINE Land Cover nomenclature for remote sensing applications, ensuring applicability in real-world RSI classification tasks. Furthermore, hierarchical label structures are constructed for other domains (e.g., medical imaging datasets) to evaluate the generalization and transferability of the proposed methods to analyzing images in other domains, i.e., beyond RSI.

**C5** Comprehensive empirical evaluation of the developed methods, including comparisons with state-of-the-art supervised and semi-supervised SOP approaches. The evaluation assesses predictive performance, computational efficiency, and hierarchical consistency, providing insights into the practical usability of the proposed methodologies.

## 1.6 Organization of the Thesis

This first section provided a general perspective and context of the thesis. The remainder is organized as follows. Chapter 2 presents the background and work related to the thesis in detail. This includes the description of SOP in image classification, particularly MLC and HMLC paradigms, architectural foundations for SOP including deep learning methods and graph neural networks, and learning frameworks under varying supervision levels from fully supervised to semi-supervised and self-supervised approaches.

Chapters 3–6 present the contributions of this thesis in detail. These chapters include papers (either published or under review) that present the proposed methods and their extensive experimental evaluations. Each chapter begins with a summary of the work presented in the paper included in the specific chapter.

Chapter 3 presents a comprehensive comparative analysis of deep learning architectures for MLC of RSIs. We explore how different CNN architectures perform when used in two distinct modes: as end-to-end classification models and as feature extractors for traditional machine learning methods. The study encompasses seven diverse RSI datasets and examines the effectiveness of various transfer learning strategies, providing insights into optimal design decisions for RSI-MLC tasks.

Chapter 4 presents two studies that extend SSL for RSI classification. In the first study, we develop an SSL framework based on PCTs and their ensembles, integrating decision tree–based learning with discriminative features extracted from CNNs. This hybrid approach improves classification performance under limited supervision by effectively combining structured learning with deep representations.

Building on this foundation, the second study investigates the computational efficiency of the proposed SSL framework by introducing dimensionality reduction techniques. Specifically, we examine how principal component analysis (PCA) can reduce the dimensionality of CNN feature spaces while preserving the framework's predictive accuracy and robustness.

Chapter 5 presents SSL-MAE, an adaptive framework that unifies self-supervised representation learning with discriminative classification to avoid the limitations of traditional two-stage SSL approaches. We propose a novel end-to-end SSL method that combines supervised and Self-SL objectives through adaptive joint learning, eliminating the computational inefficiency and suboptimal feature utilization of separate pre-training and fine-tuning phases.

Chapter 6 introduces *Hierarchical and Explicit Label Modeling* (HELM), a novel framework for HMLC that integrates graph-based hierarchical reasoning with Self-SL. We address the limitations of existing methods in handling complex multi-path hierarchies and leveraging unlabeled data by proposing a multi-token transformer architecture that captures hierarchical dependencies through graph learning.

Finally, Chapter 7 concludes by discussing the contributions to science made with this thesis and outlines directions for future work in semi-supervised SOP for remote sensing and related domains.

# Chapter 2

# Background and Related Work

This chapter establishes the theoretical and methodological foundations necessary for understanding and contextualizing the research presented in this dissertation. We first examine structured output prediction paradigms, defining multi-label and hierarchical classification tasks critical for representing complex visual content. Subsequently, we discuss learning frameworks, highlighting the spectrum of supervision strategies – from fully supervised to semi- and self-supervised approaches – that enable effective learning in scenarios with varying data annotation availability. This background sets the stage for our proposed methodological advancements and empirical evaluations detailed in subsequent chapters.

## 2.1 Structured Output Prediction in Image Classification

Image classification tasks encompass a range of prediction problems that differ in output complexity and structural constraints. While early approaches focused on assigning a single categorical label to an image, modern applications frequently demand more sophisticated output structures capable of representing the rich, multi-faceted nature of visual content. In this section, we chart the progression from simple categorical classification to structured output tasks, with specific focus on multi-label classification and hierarchical multi-label classification, the core paradigms in this dissertation [5], [6].

As illustrated in Figure 2.1, the progression from multi-class to multi-label and hierarchical multi-label reflects both an increase in output complexity and the growing need to model the natural structure of semantic concepts in images. Multi-class classification restricts predictions to exactly one class, whereas multi-label classification enables simultaneous prediction of multiple labels. Hierarchical multi-label classification introduces taxonomic relationships among labels, enforcing that predictions conform to domain hierarchies.

### 2.1.1 The task of multi-label image classification

*Multi-label classification* (MLC) generalizes the conventional single-label setting by allowing each example to be associated with multiple semantic concepts [5], [40]. This formulation is essential for domains such as remote sensing, where an image may simultaneously depict *urban infrastructure*, *vegetation*, *water bodies*, and other land-cover types [1].

Formally, the task is defined as follows:

**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$ where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).

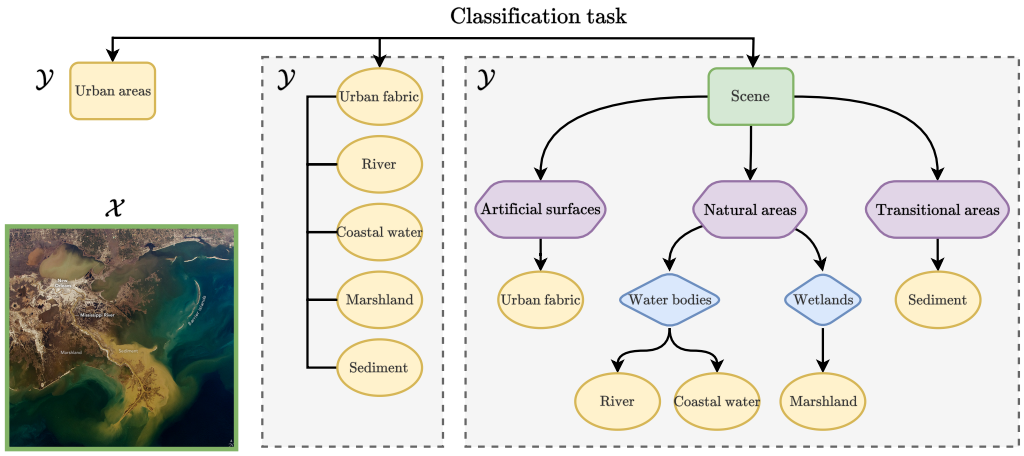



Figure 2.1: Evolution of image classification paradigms using a remote sensing image as example. Left: Multi-class classification assigns a single dominant label. Center: MLC predicts multiple co-occurring labels. Right: HMLC arranges outputs within a taxonomy, where predicting a leaf label entails predicting all its ancestors, ensuring logical consistency in the label space.

- A target space $\mathcal{Y}$ defined by a finite set of classes $C = \{c_1, c_2, \ldots, c_L\}$.
- A set of examples $E = \{(x_i, Y_i) \mid x_i \in \mathcal{X}, Y_i \subseteq C, 1 \leq i \leq N\}$, where each example $e_i \in E$ is a pair of an image $x_i$ from the descriptive space and a subset $Y_i$ from the target space, and $N$ is the number of examples.
- A quality criterion $q$ (e.g., a metric that rewards high predictive accuracy or low predictive error).

**Find:** A function

$$f : \mathcal{X} \to 2^C$$

(where $2^C$ is the power set of $C$) such that $f$ maximizes $q$.

The principal challenge in MLC is capturing and exploiting *label correlations* – dependencies between labels such as "*water*" and "*wetlands*" or "*urban*" and "*transportation*" that frequently co-occur in real-world imagery.

### 2.1.2 The task of hierarchical multi-label image classification

*Hierarchical multi-label classification* (HMLC) extends MLC by introducing *hierarchical constraints* that reflect structured relationships among labels, typically defined by a taxonomy or ontology [6], [7]. In such settings, labels are organized in a tree or directed acyclic graph (DAG), and membership in a specific class implies membership in all of its ancestor classes.

Formally, the task is defined as follows:

**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$ where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).

- A target space $\mathcal{Y}$ defined with a hierarchy of classes $(C, \leq_h)$, where $C$ is a set of classes and $\leq_h$ is a partial order among them, representing the superclass relationship, i.e., $\forall c_1, c_2 \in C, c_1 \leq_h c_2$ if and only if $c_1$ is a superclass of $c_2$.
- A set of examples $E = \{(x_i, Y_i) \mid x_i \in \mathcal{X}, Y_i \subseteq C, 1 \leq i \leq N\}$, where each example $e_i \in E$ is a pair of an image $x_i$ from the descriptive space and a subset $Y_i$ from the target space, and each subset satisfies the hierarchy constraint, i.e., $c \in Y_i, c' \in C, c' \leq_h c \Rightarrow c' \in Y_i$, and $N$ is the number of examples.
- A quality criterion $q$ (e.g., hierarchical accuracy, hierarchical F-measure).

**Find:** A function

$$f : \mathcal{X} \to 2^C$$

(where $2^C$ is the power set of $C$) such that $f$ maximizes $q$ and the predictions made by $f$ satisfy the hierarchy constraint, i.e., $c \in f(x), c' \in C, c' \leq_h c \Rightarrow c' \in f(x)$.

## 2.2 Deep Learning for SOP

This section describes the main architectural components and guiding principles that serve as the foundation for SOP methods developed in this thesis. Given the complexity of multi-label and hierarchical classification in image domains, especially when labeled data is limited, these architectures provide essential mechanisms for capturing rich visual information and modeling complex label dependencies.

Throughout this thesis, the learned visual representations are denoted as $\mathbf{z} \in \mathbb{R}^d$, where $\mathbf{z} = f_\theta(\mathbf{x})$ represents the transformation of input image $\mathbf{x}$ by a feature extractor $f_\theta$ parameterized by $\theta$. These embeddings serve as the input for subsequent structured prediction modules.

### 2.2.1 Feature extractors

Extracting meaningful features from images is a critical prerequisite for effective structured output prediction. The learned representations must capture both local patterns and global context to support correlated multi-label classes.

Two dominant architectures studied in this work are *convolutional neural networks* and *vision transformers*, distinguished by their underlying inductive biases and mechanisms for capturing spatial dependencies.

#### 2.2.1.1 Convolutional neural networks

*Convolutional Neural Networks* (CNNs) have historically formed the backbone of advances in visual recognition by exploiting two fundamental architectural principles: *spatial locality* and *translation equivariance* [3], [41]. The principle of spatial locality assumes that nearby pixels are more strongly correlated than distant ones, enabling the network to learn local feature detectors that operate on small spatial neighborhoods. Translation equivariance ensures that the same feature detector can recognize patterns regardless of their position in the input image, a property that emerges naturally from the weight-sharing mechanism inherent in convolution operations.

The core operation in a CNN is the two-dimensional convolution between an input feature map $\mathbf{x}$ and a learnable kernel (or *filter*) $\mathbf{w} \in \mathbb{R}^{k \times k}$, defined as:

$$(\mathbf{x} * \mathbf{w})[i, j] = \sum_{u,v} \mathbf{x}[i + u, j + v] \cdot \mathbf{w}[u, v], \tag{2.1}$$

where $k$ denotes the kernel size and $(i, j)$ indexes the spatial location in the output feature map. In practice, this operation corresponds to a *cross-correlation* rather than a strict mathematical convolution because the kernel is not flipped before being applied. This distinction is inconsequential for learning since the kernel weights are optimized during training. Using cross-correlation simplifies implementation and yields the same representational capacity once the parameters are learned.

Each kernel acts as a localized feature detector whose weights $\mathbf{w}[u, v]$ are tuned to respond maximally to specific visual patterns such as edges, corners, or textures. The resulting feature maps preserve the spatial arrangement of detected patterns while substantially reducing the number of learnable parameters through weight sharing across all spatial locations.

Figure 2.2 illustrates a typical CNN architecture, where stacked convolutional and pooling layers progressively transform raw pixel intensities into hierarchical feature representations. This hierarchical organization follows a compositional principle: early layers capture low-level features such as oriented edges, color gradients, and simple textures within small receptive fields (typically 3×3 or 5×5 pixels). As information propagates through deeper layers, the effective receptive field expands exponentially, enabling the network to integrate increasingly complex spatial relationships. Intermediate layers combine simple edge responses into more abstract motifs such as corners or junctions, while deeper layers encode high-level semantic concepts like object parts, faces, or entire scene categories.

The expanding receptive field is particularly important for multi-label visual recognition tasks, where understanding global context and spatial relations between multiple objects becomes essential. Pooling operations, typically max or average pooling, serve two primary roles: they introduce a degree of translation invariance by summarizing activations within local neighborhoods, and they reduce computational complexity by downsampling the spatial resolution of feature maps. However, this reduction must be balanced against the need to preserve fine-grained spatial detail required for accurate localization in multi-label or dense prediction settings.

The success of CNNs in visual recognition can be attributed to their strong inductive biases, which align well with the statistical regularities of natural images. The assumption of local connectivity dramatically reduces the parameter space from $O(n^4)$ for fully connected layers to $O(k^2 \cdot C_{\text{in}} \cdot C_{\text{out}})$, where $k$ is the kernel size, and $C_{\text{in}}$ and $C_{\text{out}}$ denote the numbers of input and output feature channels, respectively. This reduction makes training feasible on large-scale datasets while improving generalization by limiting overfitting.

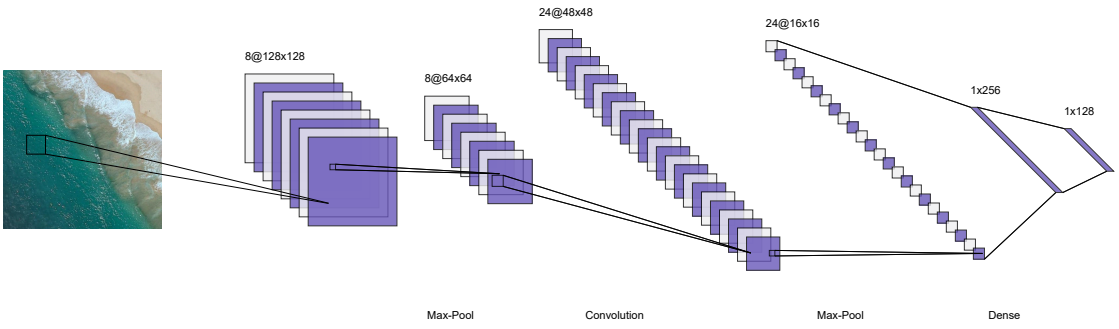



Figure 2.2: A simple CNN architecture. The network progressively transforms the input image through alternating convolution and pooling layers, extracting increasingly abstract features. The final dense layers aggregate these hierarchical features into a compact latent representation.

#### 2.2.1.2 Vision transformers

*Vision Transformers* (ViTs) propose a fundamentally different paradigm by replacing the inductive biases of CNNs with a data-driven approach based on global self-attention mechanisms [4]. Rather than assuming spatial locality, ViTs treat visual recognition as a sequence modeling problem, leveraging the transformer architecture's success in natural language processing to learn spatial relationships directly from data.

The ViT preprocessing pipeline begins by partitioning an input image of size $H \times W \times C_{\text{in}}$ into a sequence of $N_p = \frac{HW}{P^2}$ non-overlapping patches of resolution $P \times P$, where $P$ is typically 16 or 32 pixels. Each patch is flattened into a vector of dimension $P^2 \cdot C_{\text{in}}$ and linearly projected to the model's hidden dimension $D$ through a learnable embedding matrix $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C_{\text{in}}) \times D}$. This patch embedding process can be expressed as:

$$\mathbf{z}_0 = [\mathbf{x}_{\text{cls}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \ldots; \mathbf{x}_p^{N_p}\mathbf{E}] + \mathbf{E}_{\text{pos}}, \tag{2.2}$$

where $\mathbf{x}_{\text{cls}}$ is a learnable classification token prepended to the sequence, $\mathbf{x}_p^i$ represents the $i$-th flattened patch, and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N_p+1) \times D}$ are learnable positional embeddings that encode spatial structure information lost during flattening.

The ViT encoder, illustrated in Figure 2.3, processes these patch embeddings through $N_L$ identical transformer layers, each consisting of multi-head self-attention (MSA) followed by a position-wise multilayer perceptron (MLP). The multi-head self-attention mechanism enables each patch to attend to all other patches simultaneously, computing attention weights based on content similarity rather than spatial proximity. The self-attention operation is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.3}$$

where $\mathbf{Q} = \mathbf{Z}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{Z}\mathbf{W}^K$, and $\mathbf{V} = \mathbf{Z}\mathbf{W}^V$ are the queries, keys, and values derived from the input embeddings $\mathbf{Z}$ through learned projection matrices, and $d_k = D/h$ denotes the dimension of each attention head in a configuration with $h$ heads.

The multi-head attention mechanism partitions the embedding space into $h$ parallel attention heads, each learning to focus on different types of relationships:

$$\text{MSA}(\mathbf{Z}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}^O, \tag{2.4}$$

where $\text{head}_i = \text{Attention}(\mathbf{Z}\mathbf{W}_i^Q, \mathbf{Z}\mathbf{W}_i^K, \mathbf{Z}\mathbf{W}_i^V)$ and $\mathbf{W}^O$ is a learned output projection matrix.

This global attention mechanism represents a paradigm shift from CNNs' hierarchical feature learning. While CNNs build complex representations through progressive spatial aggregation, ViTs enable direct interaction among all image patches from the first layer. Each patch can attend to any other patch with weights determined by semantic similarity, allowing the model to capture long-range dependencies and complex spatial relationships that may span the entire image. This capability is particularly valuable for multi-label recognition tasks, where objects of interest may be spatially distant yet semantically related.

Because the transformer architecture is permutation invariant, explicit positional encoding is required to maintain spatial awareness. Unlike CNNs, where spatial relationships are implicitly encoded through the convolution operation, ViTs must learn them through positional embeddings $\mathbf{E}_{\text{pos}}$. Several studies have shown that ViTs can learn meaningful spatial representations, with early-layer attention maps resembling edge or contour detectors and deeper layers capturing object-part or region-level interactions [4], [42], [43].

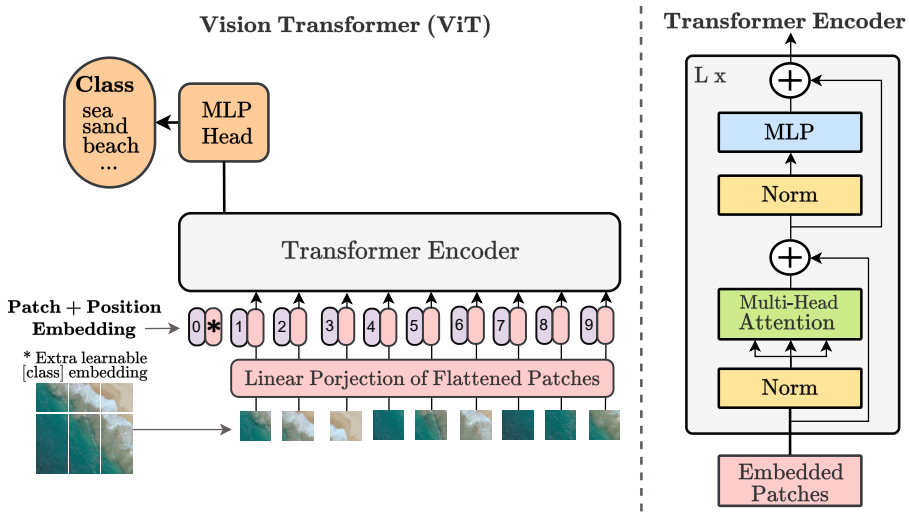



Figure 2.3: Vision Transformer architecture. An input image is divided into fixed-size patches, which are linearly embedded and combined with positional encodings. These patch embeddings are processed by a transformer encoder consisting of alternating multi-head attention and MLP blocks. The architecture enables global interactions between all patches from the first layer, with the final representation used for classification tasks.

Although ViTs offer architectural flexibility and interpretability, this comes at a computational cost. The self-attention operation scales quadratically with the number of patches, that is $O(N_p^2)$, making ViTs computationally expensive for high-resolution images. Moreover, the absence of strong inductive biases means that ViTs typically require larger datasets and longer training schedules for effective convergence. However, they exhibit excellent scaling behavior and benefit significantly from large-scale pretraining [44]–[46], often surpassing convolutional architectures when sufficient data and computational resources are available.

### 2.2.2 Graph neural networks

While visual embeddings effectively encode semantic content, explicitly modeling complex inter-label dependencies is crucial for consistent multi-label predictions. These dependencies are often hierarchical, reflecting taxonomic relationships among labels where fine-grained categories are nested within broader conceptual groups. *Graph Neural Networks* (GNNs) provide a principled framework to incorporate such structured dependencies directly into the learning process, moving beyond traditional approaches that treat labels as independent entities [38], [47]–[50].

The fundamental premise of applying GNNs to structured prediction lies in representing label relationships as explicit graph structures. In *hierarchical multi-label classification*, this translates to encoding taxonomic relationships as a directed graph where nodes represent labels and edges capture parent–child dependencies. This graph-based representation enables the propagation of information along taxonomic pathways, ensuring that predictions respect both local label correlations and global hierarchical constraints.

Formally, let the underlying class hierarchy be denoted as $\mathcal{H}$, and let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be its corresponding directed graph representation. The vertex set $\mathcal{V}$ corresponds to the class set $C$, where each vertex $v \in \mathcal{V}$ represents a class $c \in C$, and edges $\mathcal{E}$ encode parent–child relations defined by the partial order $\leq_h$. A GNN refines initial node embeddings through iterative message passing [49], [50], enabling each label to incorporate contextual informa-

tion from its taxonomic neighbors and producing embeddings that reflect both individual label semantics and structural dependencies within the hierarchy.

The message passing mechanism operates through a two-stage process at each layer. Each node $v \in \mathcal{V}$ maintains a learned embedding $\mathbf{h}_v^{(k)}$ that is updated at the $k$-th layer. First, each node computes messages to send to its neighbors:

$$\mathbf{m}_{u \to v}^{(k)} = \psi(\mathbf{h}_u^{(k-1)}, \mathbf{h}_v^{(k-1)}), \tag{2.5}$$

where $\psi$ is a learnable message function (e.g., a small neural network) that encodes relevant information transmitted between connected nodes $u$ and $v$. This function determines what information is shared between taxonomically related labels.

Second, each node aggregates incoming messages from its neighborhood $\mathcal{N}(v)$ to update its representation:

$$\mathbf{h}_v^{(k)} = \phi\left(\mathbf{h}_v^{(k-1)}, \text{AGG}\left(\left\{\mathbf{m}_{u \to v}^{(k)} : u \in \mathcal{N}(v)\right\}\right)\right), \tag{2.6}$$

where $\phi$ is an update function and $\text{AGG}(\cdot)$ denotes a permutation-invariant aggregation function such as summation, mean, or attention pooling [49].

The initialization of node embeddings serves as the foundation for this iterative refinement process. In vision tasks, these initial embeddings typically derive from the visual encoder outputs, providing each label node with semantic context from the input image. This initialization strategy ensures that message passing begins with meaningful representations that can be progressively refined through structural interactions.

The hierarchy $\mathcal{H}$ guides embedding refinement through iterative message passing [27], [49]. At each layer, nodes aggregate information from their neighbors according to parent–child relationships in $\mathcal{G}$. Parent nodes integrate specific details from their children, while child nodes incorporate broader categorical context from their parents. This bidirectional flow ensures that final embeddings capture both fine-grained semantics and broader categorical context, inherently reflecting label taxonomy and dependencies while enforcing consistency during training and prediction.

Building upon this foundation, different GNN variants employ specialized strategies for message passing and aggregation. Graph Convolutional Networks (GCNs) extend convolution operations to graph-structured data, enabling information propagation through learnable layers based on neighborhood features [38]. GraphSAGE introduces scalable sampling of fixed-size neighborhoods, learning transferable aggregation patterns that generalize to unseen nodes [48]. Graph Attention Networks (GATs) incorporate attention mechanisms that allow nodes to assign differential importance to neighbors through adaptive weights based on feature similarity [47]. Each approach offers distinct advantages for different aspects of hierarchical modeling, from computational efficiency to adaptive importance weighting.

Figure 2.4 illustrates this end-to-end process. The visual encoder $f_\theta$ processes the input image $\mathbf{x}_i$ to generate visual features $\mathbf{z}$, which initialize the label embeddings in the graph structure $\mathcal{G}$. Through multiple graph convolutional layers (typically $N_L$ layers), each message-passing step incorporates contextual information from neighboring labels according to the hierarchical structure, producing hierarchy-aware embeddings $\{\mathbf{h}_v^{(N_L)}\}_{v \in \mathcal{V}}$ that respect structural dependencies [27], [49]. These refined embeddings eliminate the need for heuristic post-processing steps to maintain label consistency, as structural constraints are naturally enforced through the graph-based learning process.

The GNN framework's compatibility with *semi-supervised learning* paradigms represents a particularly valuable property for practical applications. Information can propagate through the graph structure even when supervision is incomplete or sparse, as message passing facilitates knowledge transfer between labeled and unlabeled components [38],

[49]. This capability enables effective leveraging of partial annotations, where labels may be missing for certain nodes or training examples. The message-passing mechanism allows supervised information to flow from labeled to unlabeled nodes through the graph connections, effectively performing transductive inference that aligns naturally with the semi-supervised objectives central to this dissertation [27], [38], [49].

This property is particularly valuable in multi-label scenarios where obtaining complete annotations for all possible labels is often impractical or prohibitively expensive. By leveraging the structural relationships encoded in the hierarchy, GNNs can make informed predictions for unlabeled components based on the taxonomic context provided by labeled neighbors, significantly reducing the annotation burden while maintaining prediction quality.

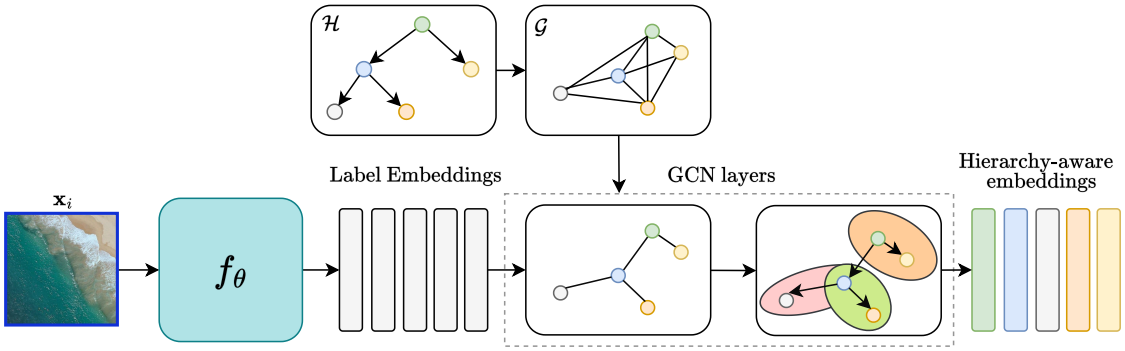



Figure 2.4: Graph Neural Network pipeline for hierarchical multi-label classification. The underlying label hierarchy $\mathcal{H}$ (left) is represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes correspond to labels and edges encode parent–child relationships. Initial node embeddings derived from the visual encoder $f_\theta$ are iteratively refined through multiple graph convolutional layers, producing hierarchy-aware embeddings that respect structural dependencies. This structured embedding process ensures semantically and hierarchically consistent multi-label predictions.

## 2.3 Learning with Varying Levels of Supervision

The effectiveness of machine learning models is fundamentally tied to the nature and extent of supervision provided during the training process. In image classification, supervision refers to the availability of ground truth labels that guide learning algorithms toward optimal parameter configurations. This section examines three distinct supervision paradigms – *supervised*, *semi-supervised*, and *self-supervised* learning – each offering unique approaches to leveraging labeled and unlabeled data with distinct characteristics, advantages, and limitations for visual recognition tasks [3], [10], [23], [51].

As illustrated in Figure 2.5, the supervision spectrum ranges from fully supervised learning with complete label information for all training examples [3], through semi-supervised learning combining limited labeled data with abundant unlabeled samples [10], to *self-supervised learning* that creates supervisory signals from data itself without manual annotations [23], [51]. Each paradigm addresses different practical constraints and data availability scenarios, offering distinct trade-offs in computational efficiency, annotation requirements, and generalization capabilities. Understanding these paradigms is crucial for developing effective learning strategies, particularly in domains where labeled data is scarce or expensive to obtain, such as remote sensing and medical imaging applications.

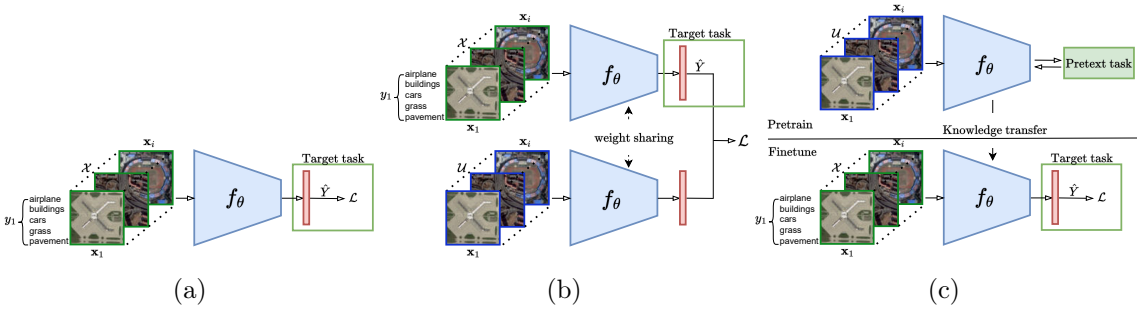



Figure 2.5: Comparison of different learning paradigms: (a) Supervised learning uses only labeled data, (b) Semi-supervised learning combines labeled and unlabeled data, (c) Self-supervised learning uses pretext tasks for knowledge transfer to target tasks.

### 2.3.1 Supervised learning

Supervised learning has achieved remarkable success in image classification, particularly through deep convolutional neural networks trained on large-scale datasets such as ImageNet [3]. The classical supervised learning paradigm in computer vision involves training a model from a dataset of labeled input-output pairs [52], [53].

Formally, the supervised learning task for structured output prediction is defined as follows:

**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$ where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).
- A target space $\mathcal{Y}$ defined by a finite set of classes $C = \{c_1, c_2, \ldots, c_L\}$.
- A training dataset $\mathcal{D} = \{(x_i, Y_i) \mid x_i \in \mathcal{X}, Y_i \subseteq C, 1 \leq i \leq N\}$, where each example consists of an input image $x_i$ from the descriptive space and a ground-truth label subset $Y_i$ from the target space, and $N$ is the number of training examples.
- A loss function $\mathcal{L} : 2^C \times 2^C \to \mathbb{R}$ that measures the discrepancy between predicted and ground-truth label subsets.
- A regularization term $\mathcal{R} : \Theta \to \mathbb{R}$ that penalizes model complexity to prevent overfitting.

**Find:** A parameterized function

$$f_\theta : \mathcal{X} \to 2^C$$

with parameters $\theta \in \Theta$ that minimizes the empirical risk:

$$\theta^* = \arg\min_{\theta \in \Theta} \left[ \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f_\theta(x_i), Y_i) + \lambda_r \mathcal{R}(\theta) \right]$$

where $\lambda_r \geq 0$ is a regularization hyperparameter.

The specific formulation varies depending on the classification paradigm:

**MCC:** Each input $x_i$ is associated with exactly one class, i.e., $|Y_i| = 1$ for all $i$. The loss function is typically categorical cross-entropy (CE).

**MLC:** Each input $x_i$ can be associated with multiple classes simultaneously, i.e., $Y_i \subseteq C$ with no cardinality restrictions. The loss function is commonly binary cross-entropy (BCE) applied independently to each class.
**HMLC:** Label subsets must additionally satisfy the hierarchy constraint defined by the partial order $\leq_h$ on $C$:

$$\forall Y_i \in \mathcal{D}: \quad c \in Y_i, c' \in C, c' \leq_h c \Rightarrow c' \in Y_i$$

The loss function incorporates hierarchical consistency, often through specialized hierarchical losses that penalize violations of parent-child relationships.
**Optimization:** The parameters $\theta^*$ are typically obtained through gradient-based optimization methods (e.g., stochastic gradient descent (SGD)), iteratively updating parameters based on gradients computed via backpropagation:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \left[ \frac{1}{|\mathcal{B}|} \sum_{(x_i, Y_i) \in \mathcal{B}} \mathcal{L}(f_{\theta^{(t)}}(x_i), Y_i) + \lambda_r \mathcal{R}(\theta^{(t)}) \right],$$

where $\eta$ is the learning rate, $\mathcal{B}$ is a mini-batch sampled from $\mathcal{D}$, and $t$ denotes the iteration number.

Despite its remarkable success, the supervised paradigm is fundamentally constrained by its reliance on comprehensive manual annotations. As model complexity increases, the number of labeled examples required to achieve good generalization also grows rapidly. In other words, more expressive hypothesis classes demand proportionally larger and more diverse training datasets to avoid overfitting and achieve reliable generalization [54], [55]. This dependence on large labeled datasets makes purely supervised learning impractical in domains where manual annotation is expensive or infeasible, such as remote sensing or medical imaging.

### 2.3.2 Semi-supervised learning

*Semi-supervised learning* (SSL) is a learning paradigm that leverages both labeled and unlabeled data to improve model generalization while mitigating the practical limitations of large-scale annotation. As illustrated in Figure 2.5(b), SSL bridges the gap between fully supervised and unsupervised learning by exploiting the abundance of unlabeled data to complement the limited set of annotated examples.

While labeled data provides explicit supervisory signals, unlabeled samples encode valuable structural and semantic information about the underlying data distribution. When appropriately integrated, this additional information helps regularize the learning process, improve feature representations, and reduce overfitting [9], [10], [51].

Formally, the SSL task for structured output prediction (SOP) can be defined as follows:
**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$, where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).
- A target space $\mathcal{Y}$ defined over a finite set of classes $C = \{c_1, c_2, \ldots, c_L\}$.
- A labeled dataset $\mathcal{D}_L = \{(x_i, Y_i) \mid x_i \in \mathcal{X}, Y_i \subseteq C, 1 \leq i \leq N_L\}$ containing $N_L$ examples with ground-truth annotations.
- An unlabeled dataset $\mathcal{D}_U = \{x_j \mid x_j \in \mathcal{X}, 1 \leq j \leq N_U\}$ containing $N_U$ examples without annotations, where typically $N_L \ll N_U$.

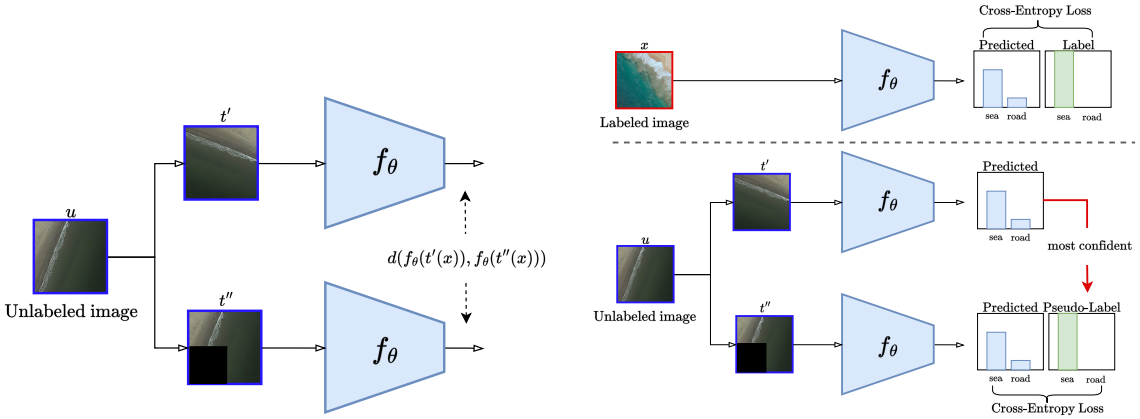
**Find:** A parameterized function

$$f_\theta : \mathcal{X} \to 2^C,$$

with parameters $\theta \in \Theta$, that minimizes the combined empirical risk:

$$\theta^* = \arg\min_{\theta \in \Theta} \left[ \frac{1}{N_L} \sum_{i=1}^{N_L} \mathcal{L}_s\big(f_\theta(x_i), Y_i\big) + \lambda_u \frac{1}{N_U} \sum_{j=1}^{N_U} \mathcal{L}_u\big(f_\theta, x_j\big) + \lambda_r \mathcal{R}(\theta) \right],$$

where $\mathcal{L}_s : 2^C \times 2^C \to \mathbb{R}$ is a supervised loss measuring prediction quality on labeled data, $\mathcal{L}_u$ is an unsupervised loss that promotes consistency or other desirable properties on unlabeled data, $\lambda_u \geq 0$ controls the contribution of the unlabeled component, $\lambda_r \geq 0$ specifies the regularization strength, and $\mathcal{R}(\theta)$ denotes a regularization term penalizing model complexity or overfitting.




(a) Consistency regularization (b) Pseudo-labeling

Figure 2.6: Core SSL paradigms. (a) Consistency regularization enforces prediction stability under different perturbations or augmentations of the same input. (b) Pseudo-labeling assigns high-confidence model predictions as provisional labels for unlabeled data, enabling iterative self-training.

The theoretical foundation of SSL rests on several key assumptions about the relationship between labeled and unlabeled data. The *cluster assumption* suggests that data points in the same cluster are likely to share similar label subsets, implying that decision boundaries should pass through low-density regions. The *manifold assumption* posits that high-dimensional data lies on a lower-dimensional manifold, where nearby points on the manifold should have similar label subsets. The *smoothness assumption* requires that the learned function should vary smoothly across the input space, ensuring consistent predictions for similar inputs [9].

Contemporary SSL encompasses three principal methodological approaches that have proven effective across different scenarios: consistency regularization, pseudo-labeling, and entropy minimization [10], [51]. Modern frameworks often combine these strategies to achieve robust performance across diverse datasets and tasks.

**Optimization:** The parameters $\theta^*$ are typically obtained through gradient-based optimization methods that jointly leverage both labeled and unlabeled data. The gradient update rule incorporates contributions from both datasets:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \left[ \frac{1}{|\mathcal{B}_L|} \sum_{(x_i, Y_i) \in \mathcal{B}_L} \mathcal{L}_s(f_{\theta^{(t)}}(x_i), Y_i) + \lambda_u \frac{1}{|\mathcal{B}_U|} \sum_{x_j \in \mathcal{B}_U} \mathcal{L}_u(f_{\theta^{(t)}}, x_j) + \lambda_r \mathcal{R}(\theta^{(t)}) \right]$$

where $\eta$ is the learning rate, $\mathcal{B}_L$ and $\mathcal{B}_U$ are mini-batches sampled from $\mathcal{D}_L$ and $\mathcal{D}_U$ respectively, and $t$ denotes the iteration number.

**Common Scheduling Strategies:** Balancing supervised and unsupervised losses often requires careful tuning or adaptive scheduling to ensure stable optimization and effective use of unlabeled data [11], [56]–[59]. Several strategies have been proposed in the literature:

- *Ramp-up scheduling:* $\lambda_u(t) = \lambda_{u,\max} \cdot w(t)$, where $w(t)$ is a monotonically increasing function (e.g., Gaussian or sigmoid) that gradually raises the contribution of the unsupervised loss from 0 to 1. This allows the model to first stabilize on labeled data before incorporating unlabeled consistency constraints [11], [56].

- *Confidence-based weighting:* Dynamically adjusting $\lambda_u$ based on the model's confidence or agreement on pseudo-labels, so that only high-confidence unlabeled predictions contribute strongly to the loss [57], [59].

- *Curriculum scheduling:* Introducing unlabeled samples in an easy-to-hard order, progressively including more uncertain or diverse examples as training progresses [60], [61].

- *Class-balanced scheduling:* Modulating $\lambda_u$ across classes to mitigate imbalance between labeled and unlabeled data distributions, ensuring that rare classes are not overwhelmed by dominant ones [62], [63].

The following subsections examine specific methodological approaches and their applications across different classification paradigms, ranging from single-label to complex hierarchical multi-label settings.

#### 2.3.2.1 Consistency regularization approaches

Consistency regularization represents one of the most successful paradigms in SSL, operating on the fundamental principle that model predictions should remain stable under reasonable perturbations or transformations of the input data [11], [56], [57], [64]. This approach leverages the assumption that semantic content should remain invariant under data augmentations, making it particularly suitable for scenarios where robust feature representations are essential.

The consistency regularization objective enforces prediction stability under different augmentation views of the same input, formally expressed as:

$$\mathcal{L}_{consistency} = \frac{1}{N_U} \sum_{j=1}^{N_U} \text{dist}(f_\theta(t(x_j)), f_\theta(t'(x_j))) \tag{2.7}$$

where $t(\cdot)$ and $t'(\cdot)$ represent different transformation strategies, and $\text{dist}(\cdot, \cdot)$ measures prediction distance, commonly implemented using Euclidean distance, BCE, or Kullback–Leibler (KL) divergence depending on the specific task requirements.

Notable methods in this paradigm include Π-Model [64], Temporal Ensembling [56], and Mean Teacher [11], which employ dropout-based stochastic transformations, ensemble prediction strategies, and teacher-student architectures with exponential moving average weights, respectively. These approaches have demonstrated exceptional effectiveness in scenarios where augmentation strategies can be carefully designed to preserve semantic content while introducing sufficient variation to regularize the learning process.

#### 2.3.2.2 Pseudo-labeling and self-training approaches

Pseudo-labeling is one of the most widely adopted strategies in SSL, where high-confidence model predictions on unlabeled data are used as surrogate supervision signals [12], [57]–[59], [65]. This self-training paradigm iteratively refines the model by treating confident predictions as provisional ground truth, thereby enlarging the effective labeled set and enabling progressive performance improvements through bootstrapped supervision.

The core pseudo-labeling mechanism operates through confidence-based label assignment. For an unlabeled sample $x_j$, the model generates predictions, and pseudo-labels are assigned when the model's confidence exceeds a predefined threshold. For SOP, this can be formalized as:

$$\hat{Y}_j = \begin{cases} \{c \in C : P(c \in f_\theta(x_j)) > \tau_c\}, & \text{if confidence criteria met,} \\ \emptyset, & \text{otherwise,} \end{cases} \tag{2.8}$$

where $P(c \in f_\theta(x_j))$ denotes the predicted probability for class $c \in C$, $\tau_c$ is the class-specific confidence threshold, and $\hat{Y}_j \subseteq C$ represents the assigned pseudo-label subset.

While conceptually simple, pseudo-labeling introduces several well-known challenges. Threshold management plays a critical role: permissive thresholds may introduce erroneous pseudo-labels that reinforce model biases and propagate noise, whereas overly conservative thresholds limit the contribution of unlabeled data [66], [67]. Moreover, the effectiveness of this mechanism depends on reliable confidence estimation, as deep neural networks are often poorly calibrated and may produce overconfident probabilities that do not accurately reflect prediction reliability [68], [69].

Determining appropriate thresholds remains an open research problem across both multi-class and structured output settings. Even in conventional classification, threshold selection is non-trivial, as the confidence distribution can vary substantially across classes and training stages [59], [66], [67]. In SOP, the problem becomes even more complex due to interdependencies among output components and hierarchical constraints, for which few dedicated pseudo-labeling strategies currently exist [70].

Traditional implementations typically rely on static, globally defined thresholds throughout training, but such heuristics often fail to account for evolving confidence dynamics or heterogeneous class learning rates. Classes may converge at different speeds, and the overall confidence distribution can shift as the model improves [58], [66]. Consequently, several recent works have proposed adaptive thresholding or curriculum-based variants that adjust confidence criteria dynamically during training [61], [66], [67].

The practical effectiveness of pseudo-labeling tends to depend on two main factors: the quality of the initial model and the reliability of its confidence estimates. A reasonably accurate initial model provides a stable foundation for generating pseudo-labels in early iterations, while well-calibrated confidence estimates ensure that only trustworthy predictions contribute to the augmented training set [65], [66]. These dependencies make pseudo-labeling particularly effective in domains where moderate supervision is available and where the model architecture supports meaningful uncertainty estimates.

#### 2.3.2.3 Hybrid approaches

Recent advances in SSL have demonstrated that combining multiple paradigms often yields superior performance compared to individual approaches. These hybrid methods strategically integrate consistency regularization, pseudo-labeling, entropy minimization, and data augmentation techniques to create unified frameworks that leverage complementary sources of supervisory signal [57], [59], [66], [71].

MixMatch represents a pioneering holistic approach that unifies the dominant paradigms of SSL [57]. The method combines confidence-based pseudo-labeling with consistency regularization through MixUp data augmentation [72], while simultaneously incorporating entropy minimization via label sharpening.

Building upon MixMatch's foundation, more sophisticated hybrid approaches have emerged. FixMatch [59] combines confidence-based pseudo-labeling with consistency regularization using weak and strong augmentations, employing dual-threshold strategies. FlexMatch [66] introduces curriculum pseudo-labeling, and FreeMatch [71] learns thresholds automatically.

Additionally, the development of adaptive and curriculum-based SSL strategies addresses the challenge of varying data quality and confidence levels throughout the training process [66], [71]. These approaches dynamically adjust learning strategies based on model confidence and data characteristics, offering more robust and generalizable solutions for real-world applications where data quality and availability may vary significantly across different regions or time periods.

#### 2.3.2.4 Predictive clustering trees for semi-supervised learning

Predictive Clustering Trees (PCTs) are a versatile framework for structured output prediction that naturally extends from supervised to semi-supervised settings [31], [32]. We first introduce the supervised PCT formulation before presenting its semi-supervised extension.

**Supervised PCTs** PCTs generalize classical decision trees by separating variables into three roles: *descriptive attributes* (features used to form tests in internal nodes; here, deep representations $\mathbf{z} = f_\theta(\mathbf{x}) \in \mathbb{R}^d$ with coordinates $Z_1, \ldots, Z_d$), *target attributes* (variables to be predicted), and *clustering attributes* (variables used to evaluate the split heuristic). For a given example, the formal target subset $Y \subseteq C$ is represented operationally as a multi-label binary vector $\mathbf{y} \in \{0,1\}^L$, where the $\ell$-th component is 1 if class $c_\ell \in Y$ and 0 otherwise. For classification tasks, these vectors serve as both the target and clustering attributes.

At each node containing a set of examples $E$, the tree-growing algorithm searches for a split $s$ that partitions $E$ into subsets $E_L(s)$ and $E_R(s)$. The optimal split $s^*$ is the one that maximizes the **impurity reduction** (or variance reduction):

$$\text{score}(s) = \text{imp}(E) - \frac{|E_L(s)|}{|E|}\,\text{imp}(E_L(s)) - \frac{|E_R(s)|}{|E|}\,\text{imp}(E_R(s)), \tag{2.9}$$

where $\text{imp}(\cdot)$ is the impurity of a set of examples, computed over the clustering attributes.

A common and unifying choice for the impurity function in PCTs is the intra-cluster variance, defined with respect to a task-appropriate distance metric $\text{dist}(\cdot,\cdot)$ over the target vectors:

$$\text{imp}(E) = \frac{1}{|E|}\sum_{e\in E} \text{dist}\left(\mathbf{y}(e),\, \bar{\mathbf{y}}_E\right)^2, \tag{2.10}$$

where $\mathbf{y}(e)$ is the target vector of example $e$ and $\bar{\mathbf{y}}_E$ is the component-wise mean vector (prototype) of the targets in set $E$.

**For MLC**, the distance is the standard (unweighted) Euclidean distance on $\mathbb{R}^L$. This simplifies Equation (2.10) to the mean of the per-label variances:

$$\text{imp}(E) = \frac{1}{L}\sum_{\ell=1}^{L} \text{Var}_E\big(Y_\ell\big). \tag{2.11}$$

**For HMLC**, the *depth-weighted Euclidean distance* is used to prioritize agreement at higher levels of the class hierarchy [7]:

$$\mathrm{dist}_w(\mathbf{y}, \mathbf{y}')^2 = \sum_{c \in C} w(c) \left(y_c - y'_c\right)^2, \qquad \text{where} \quad w(c) = w_0^{\mathrm{depth}(c)}, \tag{2.12}$$

and $C$ is the set of all classes. A typical value for the weighting factor is $w_0 = 0.75$. This distance $\mathrm{dist}_w$ is then used in the general impurity formulation of Equation (2.10).

**Semi-supervised PCTs** The PCT framework extends to SSL by defining the impurity as a combination of clustering signals from both the descriptive and target spaces [36]. Let the descriptive attributes be $\mathbf{Z}$ (with coordinates $Z_1, \ldots, Z_d$) and the target attributes be $\mathbf{Y}$ (with coordinates $Y_1, \ldots, Y_L$). The impurities for each space are defined as their normalized variance:

$$\mathrm{imp}_Z(E) = \frac{1}{d} \sum_{i=1}^{d} p_i^Z \, \frac{\mathrm{Var}_E(Z_i)}{\mathrm{Var}_{\mathrm{train}}(Z_i)}, \tag{2.13}$$

$$\mathrm{imp}_Y(E) = \begin{cases} \dfrac{1}{L} \displaystyle\sum_{\ell=1}^{L} p_\ell^Y \, \dfrac{\mathrm{Var}_E(Y_\ell)}{\mathrm{Var}_{\mathrm{train}}(Y_\ell)} & \text{(for MLC)} \\ \dfrac{\mathrm{imp}_{\mathrm{dist}_w}(E)}{\mathrm{imp}_{\mathrm{dist}_w}(E_{\mathrm{train}})} & \text{(for HMLC),} \end{cases} \tag{2.14}$$

where $p_i^Z, p_\ell^Y$ are optional non-negative weights (defaulting to 1), and normalization by the total training set variance ensures the two impurity measures are on a comparable scale. The term $\mathrm{imp}_{\mathrm{dist}_w}(E)$ refers to the impurity calculated using Equation (2.10) with the weighted distance $\mathrm{dist}_w$.

The final semi-supervised impurity is a convex combination of the two:

$$\mathrm{imp}_{\mathrm{SS}}(E) = (1 - \omega)\, \mathrm{imp}_Z(E) + \omega\, \mathrm{imp}_Y(E), \quad \text{where } \omega \in [0, 1]. \tag{2.15}$$

This combined impurity is then used in the score function (Equation (2.9)). The hyperparameter $\omega$ controls the level of supervision:

- $\omega = 1$ **(Fully Supervised):** Only the target-space impurity $\mathrm{imp}_Y$ guides the splits. Unlabeled data are ignored in the splitting criterion.
- $0 < \omega < 1$ **(Semi-supervised):** Both descriptive-space structure (from all examples) and target-space structure (from labeled examples) influence the splits.
- $\omega = 0$ **(Unsupervised Clustering):** Only the descriptive-space impurity $\mathrm{imp}_Z$ guides tree construction. Target labels are used only to form predictions in the leaf nodes.

This formulation allows PCTs to seamlessly leverage structural information from unlabeled data while retaining the predictive power and interpretability of decision trees.

### 2.3.3 Self-supervised learning

Self-supervised learning (Self-SL) has emerged as a transformative paradigm that addresses data scarcity by learning generalizable feature representations from unlabeled data alone [14]–[16]. Unlike SSL, this approach completely eliminates the need for labeled data

during the initial representation learning phase, instead creating synthetic supervisory signals through carefully designed pretext tasks that enable models to capture fundamental visual structures and relationships without manual annotations.

The Self-SL framework consists of two stages:

**Stage 1: Pretext Training**
**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$ where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).
- An unlabeled training dataset $\mathcal{D}_U = \{x_j \mid x_j \in \mathcal{X}, 1 \leq j \leq N_U\}$ containing $N_U$ examples without annotations.
- A pretext task function $T : \mathcal{X} \to \mathcal{X} \times \mathcal{S}$ that transforms input $x$ into $(x', s)$ where $x'$ is the transformed input and $s \in \mathcal{S}$ is the synthetic supervisory signal.
- A pretext loss function $\mathcal{L}_{\text{pretext}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ that measures the quality of pretext predictions.
- A regularization parameter $\lambda_r \geq 0$.

**Find:** An encoder function $f_\theta : \mathcal{X} \to \mathbb{R}^d$ and pretext head $h_\phi : \mathbb{R}^d \to \mathcal{S}$ with parameters $\theta$ and $\phi$ that minimize:

$$(\theta^*, \phi^*) = \arg\min_{\theta,\phi} \frac{1}{N_U} \sum_{j=1}^{N_U} \mathcal{L}_{\text{pretext}}(h_\phi(f_\theta(x'_j)), s_j) + \lambda_r \|\theta\|^2 \tag{2.16}$$

where $(x'_j, s_j) = T(x_j)$ and $\lambda_r$ controls regularization strength.

**Stage 2: Downstream Adaptation**
**Given:**

- A descriptive (input) space $\mathcal{X} \subseteq \mathbb{R}^{H \times W \times C_h}$ where each input $x \in \mathcal{X}$ is an image with height $H$, width $W$, and $C_h$ channels (e.g., RGB channels for natural images, or multispectral bands for remote sensing imagery).
- A target space $\mathcal{Y}$ defined by a finite set of classes $C = \{c_1, c_2, \ldots, c_L\}$.
- A labeled training dataset $\mathcal{D}_L = \{(x_i, Y_i) \mid x_i \in \mathcal{X}, Y_i \subseteq C, 1 \leq i \leq N_L\}$ containing $N_L$ examples with ground-truth annotations.
- A pre-trained encoder $f_{\theta^*}$ from Stage 1.
- A downstream loss function $\mathcal{L}_{\text{downstream}} : 2^C \times 2^C \to \mathbb{R}$.

**Find:** A task-specific head $g_\psi : \mathbb{R}^d \to 2^C$ with parameters $\psi$ such that the final prediction function $\hat{Y} = g_\psi(f_{\theta^*}(x))$ minimizes the downstream objective.

Two adaptation strategies are commonly employed:

1. *Linear evaluation:* Freeze the pre-trained encoder $f_{\theta^*}$ and optimize only the task head parameters $\psi$.
2. *Fine-tuning:* Jointly optimize both the encoder parameters $\theta$ (initialized from $\theta^*$) and task head parameters $\psi$.

The theoretical motivation for Self-SL stems from the observation that natural images contain rich structural patterns and relationships that can serve as supervisory signals when properly formulated. By learning to predict these inherent properties, models develop feature representations that capture fundamental visual concepts transferable across diverse tasks and domains [14]–[16], [23], [51], [73]–[76].

Contemporary self-supervised methods can be broadly categorized into three principal paradigms, each offering distinct advantages for learning visual representations. *Context-based* pretext tasks, including rotation prediction [74] (where the network learns to identify the correct orientation of images), colorization [75] (reconstructing color information from grayscale inputs), and jigsaw puzzle solving [73] (reassembling shuffled image patches), encourage models to learn spatial and semantic patterns by predicting properties of carefully designed input transformations. *Contrastive learning* frameworks, including methods such as SimCLR [14], MoCo [37], and BYOL [15], learn invariant representations by comparing different augmented views of images, maximizing agreement between positive pairs while minimizing similarity to negative examples. *Generative* approaches, particularly masked image modeling (MIM) methods such as Masked Autoencoders (MAE) [16] and Bidirectional Encoder Representations from Image Transformers (BEiT) [76], reconstruct masked portions of input images to capture fine-grained visual details and contextual relationships.

This section examines two representative methods from the most prominent paradigms: SimCLR as an exemplar of contrastive learning approaches, and MAE representing generative reconstruction methods. These methods have demonstrated exceptional effectiveness in developing transferable visual representations and have become foundational architectures in modern self-supervised learning.

#### 2.3.3.1 Contrastive learning approaches

Contrastive learning frameworks constitute one of the most successful paradigms in Self-SL, learning discriminative feature representations by comparing different augmented views of images. These methods operate on the fundamental principle that different transformations of the same image should yield similar representations in the learned feature space, while representations of different images should be well-separated.

The contrastive learning objective maximizes agreement between positive pairs (different views of the same image) while minimizing similarity to negative examples (views from different images). Notable methods in this paradigm include SimCLR [14], MoCo [37], and BYOL [15], each contributing distinct architectural innovations and training strategies. This approach assumes that augmentation-invariant features capture semantic content that transfers effectively to downstream classification tasks, making it particularly suitable for scenarios where robust feature representations are essential.

SimCLR [14] represents a seminal contribution to contrastive Self-SL, demonstrating that simple frameworks can achieve remarkable performance through careful design of data augmentation strategies and loss formulations. The method operates by creating multiple augmented views of each image in a batch, computing their embeddings through an encoder-projector architecture, and optimizing a contrastive loss that brings together representations of the same image while pushing apart representations of different images.

Algorithm 2.1 presents the complete SimCLR training procedure. The framework employs a two-network architecture consisting of an encoder $f$ that extracts visual representations and a projection head $g$ that maps representations to a space where contrastive loss is applied. For each image $\mathbf{x}_i$ in the batch, two random augmentations are sampled from a distribution $\mathcal{T}$, creating positive pairs that should have similar representations despite visual differences introduced by augmentation.

The contrastive loss computation involves calculating cosine similarities between all

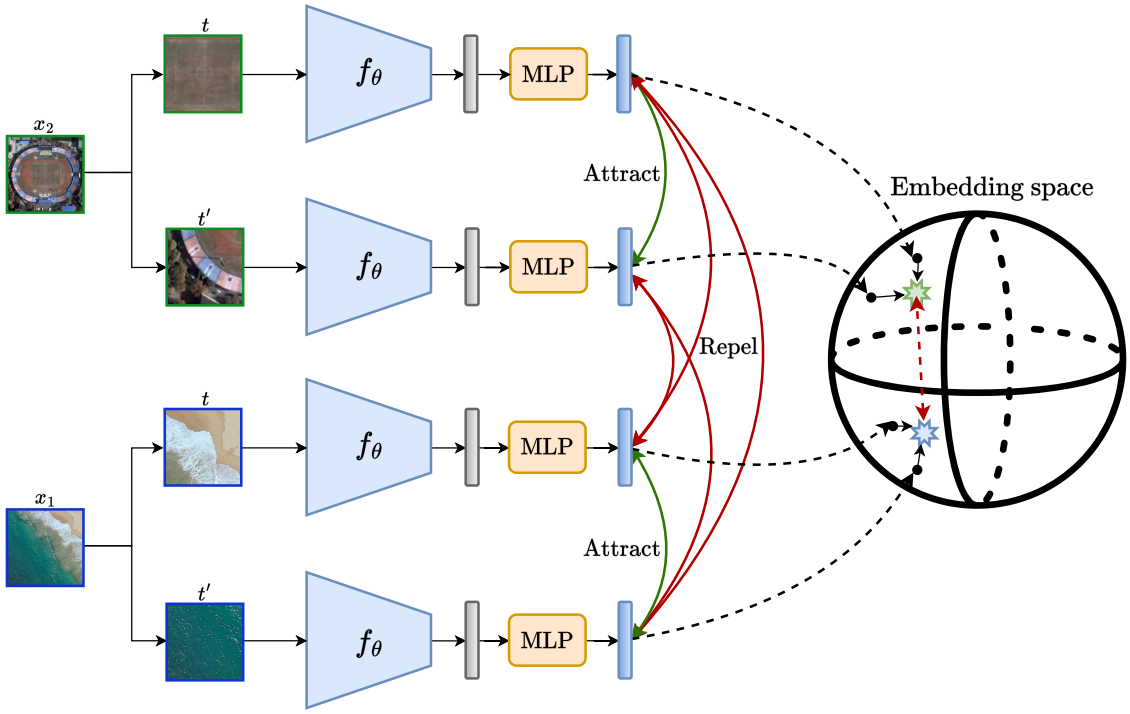



Figure 2.7: Contrastive learning framework for visual representation learning. The architecture processes pairs of images through shared visual encoders $f_\theta$ followed by projection heads (MLP) to generate normalized embeddings in a hyperspherical space. For each anchor image $x_i$, augmented views $t$ and $t'$ are created and mapped to nearby points in the embedding space. Simultaneously, embeddings from different images are pushed apart. This contrastive objective learns discriminative visual representations by maximizing agreement between semantically similar views while minimizing similarity to negative samples, resulting in a structured embedding space where semantically related images cluster together.

pairs of representations in the augmented batch, followed by a normalized temperature-scaled cross-entropy loss that encourages high similarity for positive pairs while maintaining low similarity for negative pairs. The temperature parameter $\tau$ controls the concentration of the distribution, with lower values producing more focused similarity distributions.

Upon completion of self-supervised pre-training, the projection head $g$ is discarded, and the encoder $f$ serves as a feature extractor for downstream tasks. This design reflects the observation that while the projection head facilitates contrastive learning, the encoder captures more general visual representations suitable for transfer learning applications.

#### 2.3.3.2 Generative reconstruction approaches

Generative reconstruction methods represent an alternative paradigm in Self-SL that learns representations by reconstructing corrupted or masked portions of input images. Within this category, MIM methods have gained significant popularity and pose a considerable challenge to the prevailing dominance of contrastive learning [16], [76]–[78]. MIM methods, including BEiT [76], MAE [16], Context Autoencoders (CAE) [78], and SimMIM [77], leverage co-occurrence relationships among image patches as supervision signals.

MIM represents a variant of denoising autoencoders, drawing inspiration from the success of *Masked Language Modeling* (MLM) in natural language processing, as popularized

**Algorithm 2.1:** SimCLR's main learning algorithm.

**input:** batch size $N$, temperature $\tau$, encoder $f$,
projection head $g$, augmentation set $\mathcal{T}$
**for** sampled minibatch $\{\mathbf{x}_i\}_{i=1}^{N}$ **do**
**for** all $i \in \{1, \ldots, N\}$ **do**
draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
// first augmentation
$\tilde{\mathbf{x}}_{2i-1} = t(\mathbf{x}_i)$
$\mathbf{h}_{2i-1} = f(\tilde{\mathbf{x}}_{2i-1})$ // representation
$\mathbf{z}_{2i-1} = g(\mathbf{h}_{2i-1})$ // projection
// second augmentation
$\tilde{\mathbf{x}}_{2i} = t'(\mathbf{x}_i)$
$\mathbf{h}_{2i} = f(\tilde{\mathbf{x}}_{2i})$ // representation
$\mathbf{z}_{2i} = g(\mathbf{h}_{2i})$ // projection
**end for**
**for** all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
$s_{i,j} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$ // cosine similarity
**end for**
define $\ell(i, j)$ as:
$\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp(s_{i,k}/\tau)}$
$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^{N} [\ell(2i-1, 2i) + \ell(2i, 2i-1)]$
update networks $f$ and $g$ to minimize $\mathcal{L}$
**end for**
**return** encoder network $f(\cdot)$, and discard $g(\cdot)$

by models such as *Bidirectional Encoder Representations from Transformers* (BERT) [79] and the *Generative Pre-trained Transformer* (GPT) family [76]. The introduction of ViT marked a pivotal moment that enabled effective adaptation of BERT-like pre-training strategies to computer vision [4], [16], [76]. These approaches operate on the principle that effective reconstruction requires comprehensive understanding of visual patterns, relationships, and contextual dependencies, leading to feature representations that capture both local details and global structure.

MAE [16] exemplify the generative reconstruction paradigm, demonstrating that masking large portions of input images and learning to reconstruct the missing content produces highly effective visual representations. This approach draws inspiration from the success of MLM in natural language processing, adapting the concept to visual domains through patch-based image tokenization and asymmetric encoder-decoder architectures.

The MAE framework employs a vision transformer architecture where images are divided into non-overlapping patches, and a substantial fraction (typically 75%) of patches are randomly masked during training. The encoder processes only the visible patches, creating a compact representation that must capture sufficient information for the decoder to reconstruct the entire image. This asymmetric design ensures computational efficiency while forcing the encoder to learn meaningful representations from limited visual information.

Algorithm 2.2 details the complete MAE training procedure. The method begins by patchifying input images and randomly selecting a subset of patches for masking. The encoder processes only visible patches, producing latent representations that are subsequently passed to a lightweight decoder responsible for reconstructing the complete image, includ-

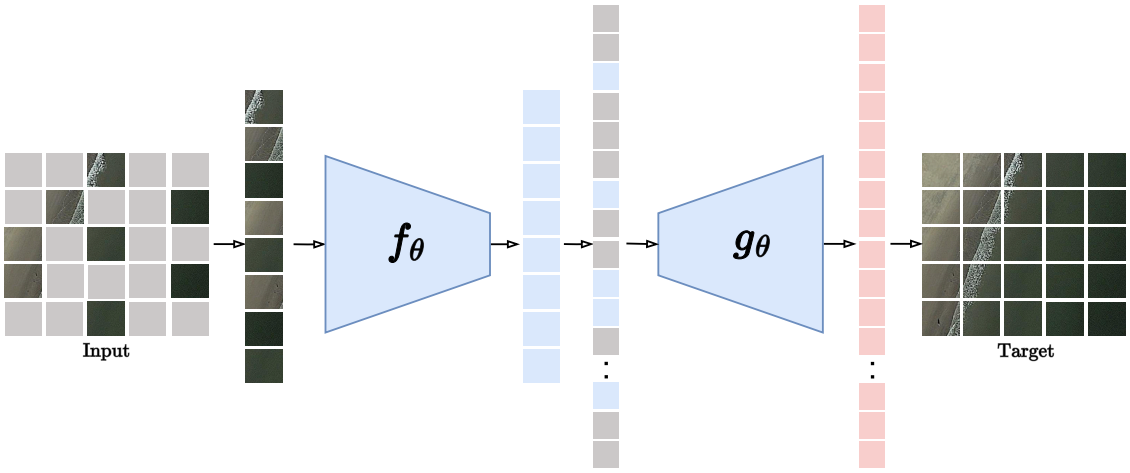



Figure 2.8: Masked Autoencoder (MAE) self-supervised pre-training framework. The input image is divided into patches and a large subset (typically 75%) is randomly masked (shown in gray). The encoder $f_\theta$ processes only the visible patches (shown in green/dark patches), creating a compact representation. The decoder $g_\theta$ then reconstructs the original image from this partial encoding, learning to predict the masked patches (shown in pink/red in the target). This asymmetric encoder-decoder architecture enables efficient pre-training by processing only a fraction of the input while learning rich visual representations through the challenging reconstruction task of predicting missing visual content.

ing masked regions. The reconstruction loss is computed exclusively on masked patches using mean squared error between original and reconstructed pixel values. This design choice ensures that the model learns to infer missing content rather than simply copying visible information, promoting the development of robust contextual understanding and feature representations.

Following pre-training, the decoder is discarded, and the encoder serves as a feature extractor for downstream applications. The high masking ratio employed during training forces the encoder to develop efficient and informative representations that can capture essential image characteristics from limited visual input, making these features particularly valuable for transfer learning scenarios.

## 2.4 HMLC of RSI with Semi-Supervised Learning

SSL holds significant promise for RSI analysis by leveraging both labeled and unlabeled data to reduce annotation demands while improving performance over purely supervised approaches [11], [13]. However, the vast majority of SSL methods for RSI classification focus exclusively on multi-class scenarios [22], [80]–[85], treating each image as belonging to a single class despite the inherently multi-label nature of many remote sensing datasets.

Multi-label ground truth annotation for RSIs demands substantially greater complexity and effort than multi-class labeling, exacerbating data scarcity and amplifying the need for SSL solutions tailored to MLC. While some methods have been extended to remote sensing contexts – including MSMatch [81] (adapting FixMatch [13] to multi-spectral data), graph-theoretic SSL [22], High-Rankness Regularized Semi-Supervised Deep Metric Learning (HR-S$^2$DML) [82], and SSL bidirectional alignment [84] – these approaches fundamentally treat labels as independent entities. This limitation prevents them from modeling label correlations or hierarchical relationships inherent in remote sensing tax-

**Algorithm 2.2:** Masked Autoencoder (MAE) Training Algorithm.

**input:** batch of images $\mathcal{B} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, mask ratio $m$, encoder $f_{enc}$, decoder $f_{dec}$
**for** each image $\mathbf{x}_i$ in $\mathcal{B}$ **do**
  # 1. Create patches from image
  $\mathcal{P}_i = \{\mathbf{p}_{i,1}, \ldots, \mathbf{p}_{i,P}\} \leftarrow \text{Patchify}(\mathbf{x}_i)$  # $\mathcal{P}_i$ is the set of $P$ patches

  # 2. Randomly mask a large portion of patches
  $P_{vis} = \lfloor P \cdot (1 - m) \rfloor$  # Number of visible patches
  $\mathcal{I}_{vis} \leftarrow \text{RandomSample}(\{1, \ldots, P\}, P_{vis})$  # Indices of visible patches
  $\mathcal{I}_{mask} \leftarrow \{1, \ldots, P\} \setminus \mathcal{I}_{vis}$  # Indices of masked patches

  # 3. Encoder processes only visible patches
  $\mathcal{P}_{vis} \leftarrow \{\mathbf{p}_{i,j} \mid j \in \mathcal{I}_{vis}\}$  # Select visible patches
  $\mathbf{z}_i \leftarrow f_{enc}(\mathcal{P}_{vis})$  # Get latent representation

  # 4. Decoder reconstructs all patches from latent representation and mask tokens
  $\hat{\mathcal{P}}_i = \{\hat{\mathbf{p}}_{i,1}, \ldots, \hat{\mathbf{p}}_{i,P}\} \leftarrow f_{dec}(\mathbf{z}_i, \mathcal{I}_{vis}, \mathcal{I}_{mask})$  # Reconstructed patches

  # 5. Calculate loss only on the masked patches
  $\mathcal{L}_i = \frac{1}{|\mathcal{I}_{mask}|} \sum_{j \in \mathcal{I}_{mask}} \|\mathbf{p}_{i,j} - \hat{\mathbf{p}}_{i,j}\|_2^2$  # Mean Squared Error
**end for**

# Compute total loss for the batch
$\mathcal{L}_{batch} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i$

# Update model weights
Update $f_{enc}$ and $f_{dec}$ via backpropagation to minimize $\mathcal{L}_{batch}$

**return** Encoder $f_{enc}$  # Discard decoder $f_{dec}$ after training

onomies, despite exploring multi-modal fusion and metric learning strategies.

Recent developments in HMLC within computer vision have produced a diverse array of methodologies, encompassing coarse-to-fine architectural designs [86], [87], hierarchy-regularized loss functions [88], [89], and contrastive or graph-based frameworks that exploit inter-class dependencies [27], [29], [49], [90]–[92]. Despite these advances, their direct application to remote sensing imagery remains largely unexplored. A notable exception is the hierarchy and exclusion (HEX) graph-based model [93], which represents one of the few successful integrations of hierarchical label constraints within a remote sensing context, specifically targeting ship classification in aerial imagery and demonstrating clear benefits of hierarchy-aware design principles. Nevertheless, this approach operates under the restrictive assumption of single-path hierarchies, where only one leaf node in the taxonomic hierarchy is active per image (i.e., exactly one ship type is present). Furthermore, no existing frameworks simultaneously address HMLC and SSL in the remote sensing domain.

In summary, while SSL has begun to alleviate annotation bottlenecks in RSI classification, the field still lacks principled SSL frameworks capable of effectively handling both multi-label and hierarchical class structures, as demanded by real-world remote sensing data. This gap is especially pronounced given the prevalence of multi-path hierarchies, strong inter-label dependencies, and severe class imbalance commonly found in RS ontologies such as CORINE Land Cover [8]. Addressing these challenges requires the development

of methods that can (i) explicitly model inter-label and hierarchical dependencies, (ii) integrate consistency regularization strategies within multi-label, hierarchy-aware settings, and (iii) unify graph-based reasoning and self-supervised pretraining within scalable, end-to-end architectures. In this dissertation, we move toward these goals by proposing hybrid frameworks that integrate semi-supervised decision-tree methodologies [36] with deep visual encoding [94], hierarchy-aware SSL objectives, and unified optimization to address the unique challenges of large-scale environmental monitoring from RSIs.

# Chapter 3

# Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images

In this chapter, we present a comprehensive comparative analysis of deep learning architectures for MLC of RSI. We explore how different CNN architectures perform when used in two distinct modes: as end-to-end classification models and as feature extractors for traditional ML methods. The study encompasses seven diverse RSI datasets and examines the effectiveness of various transfer learning strategies, from ImageNet linear evaluation to full fine-tuning approaches. We also investigate the impact of dataset size and characteristics on the relative performance of different architectural choices, providing insights into optimal design decisions for RSI-MLC tasks.

The motivation stems from the need to understand how different network architectures perform across diverse RSI datasets with varying characteristics, and whether the combination of deep feature extraction with tree ensemble methods can outperform pure deep learning approaches. Traditional single-label classification approaches are insufficient for describing the rich semantic content present in RSI, while the choice between end-to-end deep learning and hybrid approaches combining deep feature extraction with classical ML methods remained unexplored. The main contributions are the following:

1. We present an experimental analysis of different approaches for MLC of RSI. More precisely, we investigate the performance of several deep learning network architectures by using linear evaluation and fine-tuning as the main learning strategies for the MLC task.

2. We evaluate the effectiveness of the deep models used as end-to-end approaches to MLC, and used as feature extractors that provide feature representations of RSI, as inputs to tree ensemble methods for MLC. Moreover, we investigate which of the network architectures is the most suitable choice in terms of performance.

From the hypotheses we defined in Section 1.3, this chapter addresses the following:

**H1** CNN architectures used as feature extractors combined with tree ensemble methods can achieve competitive or superior performance compared to end-to-end CNN approaches for MLC, and lightweight network architectures can achieve better performance than heavier architectures while being more computationally efficient.

Our experimental analysis confirmed **H1** across seven diverse RSI datasets. The results demonstrated that tree ensemble methods combined with proper deep feature extractors outperformed end-to-end approaches in certain cases, particularly when using stratified sampling strategies. EfficientNets proved to be the most suitable architectural choice, achieving comparable predictive performance to heavier architectures like ResNet variants while having significantly fewer parameters, thus confirming their superior computational efficiency for RSI-MLC tasks.

The paper included in this Chapter is:

- Stoimchev, M., Kocev, D., Džeroski, S., "Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images" in *Remote Sensing*, 15(2), 538.

**The contributions of Marjan Stoimchev to this paper are as follows**. He contributed to the conceptual design of the experimental study, implemented all deep learning architectures and training procedures, conducted the comprehensive experimental evaluation across seven RSI datasets, performed the statistical analysis including Friedman and Nemenyi tests, analyzed the results in terms of learning strategies, architectural choices, and data availability effects, and created all visualizations and comparative analyses. He drafted the paper and revised it following feedback from the co-authors and the reviewers.

**Code availability.** All code developed for the experiments presented in this chapter, including model implementations, training pipelines, and evaluation scripts, is publicly available at: `https://github.com/marjanstoimchev/RSMLC/tree/master`.

*remote sensing*

MDPI

*Article*

# Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images

**Marjan Stoimchev [1,2,*], Dragi Kocev [1,2,3] and Sašo Džeroski [1,2]**

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia; dragi.kocev@ijs.si (D.K.); saso.dzeroski@ijs.si (S.D.)
[2] Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia
[3] Bias Variance Labs, 1000 Ljubljana, Slovenia
* Correspondence: marjan.stoimchev@ijs.si

**Abstract:** Data in the form of images are now generated at an unprecedented rate. A case in point is remote sensing images (RSI), now available in large-scale RSI archives, which have attracted a considerable amount of research on image classification within the remote sensing community. The basic task of single-target multi-class image classification considers the case where each image is assigned exactly one label from a predefined finite set of class labels. Recently, however, image annotations have become increasingly complex, with images labeled with several labels (instead of just one). In other words, the goal is to assign multiple semantic categories to an image, based on its high-level context. The corresponding machine learning tasks is called multi-label classification (MLC). The classification of RSI is currently predominantly addressed by deep neural network (DNN) approaches, especially convolutional neural networks (CNNs), which can be utilized as feature extractors as well as end-to-end methods. After only considering single-target classification for a long period, DNNs have recently emerged that address the task of MLC. On the other hand, trees and tree ensembles for MLC have a long tradition and are the best-performing class of MLC methods, but need predefined feature representations to operate on. In this work, we explore different strategies for model training based on the transfer learning paradigm, where we utilize different families of (pre-trained) CNN architectures, such as VGG, EfficientNet, and ResNet. The architectures are trained in an end-to-end manner and used in two different modes of operation, namely, as standalone models that directly perform the MLC task, and as feature extractors. In the latter case, the learned representations are used with tree ensemble methods for MLC, such as random forests and extremely randomized trees. We conduct an extensive experimental analysis of methods over several publicly available RSI datasets and evaluate their effectiveness in terms of standard MLC measures. Of these, ranking-based evaluation measures are most relevant, especially ranking loss. The results show that, for addressing the RSI-MLC task, it is favorable to use lightweight network architectures, such as EfficientNet-B2, which is the best performing end-to-end approach, as well as a feature extractor. Furthermore, in the datasets with a limited number of images, using traditional tree ensembles for MLC can yield better performance compared to end-to-end deep approaches.

**Keywords:** remote sensing; convolutional neural networks; tree ensemble methods; multi-label classification

check for updates

**Citation:** Stoimchev, M.; Kocev, D.; Džeroski, S. Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images. *Remote Sens.* **2023**, *15*, 538. https://doi.org/10.3390/rs15020538

Academic Editor: Michael K. Ng

Received: 17 October 2022
Revised: 16 December 2022
Accepted: 5 January 2023
Published: 16 January 2023

**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Remote sensing is the process of detecting and monitoring the physical characteristics of an area by measuring its reflected and emitted radiation at a distance. In the past few years, advances in satellite technology have resulted in large-scale remote sensing image (RSI) archives, which have attracted a considerable amount of research in various application areas. RSI can be used to monitor and predict various environmental phenomena, such as weather and climate change [1], land use and land cover changes at macro scale [2], deforestation [3], wildfires [4,5], and many others.

*Remote Sens.* **2023**, *15*, 538 

In machine learning terms, a wide range of applications stemming from RSI are approached using single-label classification, where the goal is to assign a single label/semantic category to an image [6]. However, real-world RSIs are typically complex and present more than a single semantic category within a single image. Hence, single-label classification is often insufficient to fully describe the presence of complex areas, which can carry semantically complex content [7].

To facilitate more realistic representation of the content of RSI, the analysis of RSI should be addressed through the task of multi-label classification (MLC), where a given image can be associated with multiple semantic concepts/labels taken from a predefined set of labels. In this way, the classification problem becomes more challenging as compared to single-label classification, as discussed in a recent large-scale comparative study and analysis of a wide range of MLC methods [8]. This study highlights the two major challenges that can limit the performance of MLC methods: the presence of complex label correlations and high-dimensional label spaces.

These challenges are attracting increasing amounts of attention from researchers focusing on deep learning (DL) methods, specifically on methods capable of automatically learning long-range dependencies (e.g., with the use of self-attention mechanisms in the vision transformer (ViT) base network architectures [9]), and handling the high-dimensional label spaces. Their internal mechanisms and structure, such as the hierarchical design and characteristics in convolutional neural networks (CNNs), with local connectivity and non-linearity, are capable of encoding information exceptionally well. Moreover, their flexible design offers knowledge to be extracted and discriminative representations to be learned from noisy data in an end-to-end manner, and achieves more accurate recognition performance in less-constrained environments as compared with traditional MLC approaches. Furthermore, the recent success of these methods can be associated with their ability to leverage large amounts of labeled data in order to learn meaningful knowledge.

Many of the existing approaches in computer vision try to address the challenges encountered by exploiting proven DL network architectures pre-trained on large-scale and diverse datasets such as ImageNet [10]. This is usually achieved by using the transfer learning paradigm [11], where specific parts of the model are fine-tuned in order to learn new features that generalize better to the new downstream task [12–14]. For example, Wang et al. [15] propose a framework based on a VGG-16 CNN model initialized with weights learned on ImageNet, which is used to extract semantic representations from images and is coupled with a recurrent neural network (RNN) network architecture with long short-term memory (LSTM) units to capture image/label relations and label dependency. Chen et al. [16] propose an end-to-end learning method based on Graph Convolutional Networks (GCN) to capture the label correlations, and a novel re-weighting scheme for creating the label correlation matrix that is used to guide the information propagation among the nodes in the GCN.

In addition, with the increased availability of RSI and the increased research interest in remote sensing applications, many interesting approaches have been proposed at the crossroads of remote sensing and computer vision [7,17–19]. For example, Hua et al. [20] propose a novel approach for MLC from aerial imagery—an attention aware label relational network, comprising a label-wise feature parcel module, an attention region extraction module, and a label relational inference module. Sumbul et al. [21] present a K-Branch CNN that uses a multi-attention strategy for bidirectional LSTM networks, which is specially developed to capture spatial and spectral contents from RS images of local image areas. Wang et al. [22], despite using local attention [21], are also able to maintain global context through global attention pooling, where the combination of both helps in modeling long-range dependencies among multiple objects and captures underlying relationships among multiple labels.

In this work, our main focus is on investigating the potential of several prominent deep learning architectures (variants of VGG [23], ResNet [24] and EfficientNet [25]) as feature extractors for the MLC of RSI as well as end-to-end approaches to MLC of RSI. To this

end, we evaluate the performance of the architectures across seven MLC RSI datasets with different properties in terms of number of images, number of labels, and average number of labels per image (label cardinality). More specifically, we use pre-trained network architectures with weights learned on ImageNet as an initialization procedure. We further perform calibration of the pre-trained models by fine-tuning them on a small set of RSIs and compare the fine-tuned and pre-trained MLC performance. Finally, we use the learned models as feature extractors to describe the RSI, and use the resulting feature vectors to learn tree ensembles such as random forests and extra trees for MLC.

The main contributions of this paper can be summarized as follows:

- We present an experimental analysis of different approaches for MLC of RSI. More precisely, we investigate the performance of several deep learning network architectures by using pre-training and fine-tuning as the main learning strategies for the MLC task.
- We evaluate the effectiveness of the deep models used as end-to-end approaches to MLC, and used as feature extractors that provide feature representations of RSI, as inputs to tree ensemble methods for MLC. Moreover, we investigate which of the network architectures is the most suitable choice in terms of performance.
- We also investigate the performance of the considered methods in terms of the influence of the number of labeled training examples by providing the methods with different fractions of the data.

The remainder of this paper is organized as follows. Section 2 describes the MLC RSI data (Section 2.1) and the machine learning methods (Section 2.2) used to analyze them (deep learning architectures in Section 2.2.1 and tree ensembles in Section 2.2.2). Next, Section 3 gives the experimental questions and the specific experimental setup, including parameter instantiations for the methods (Section 3.1), the evaluation strategy (Section 3.2), and the different evaluation measures used to assess the performance of the models (Section 3.3). Furthermore, Section 4 discusses the results of our experiments, focusing on the outcomes of the different representation learning strategies (Section 4.1), deep architectures (Section 4.2), MLC approaches (Section 4.3), and number of images (Section 4.4). Finally, Section 5 concludes the paper by providing a summary of the presented work and directions for further work.

## 2. Materials and Methods

### *2.1. Datasets*

We use seven publicly available MLC RSI datasets to assess the performance of the MLC methods, namely UC Merced (UCM) Land Use, AID Multilabel, Ankara HIS archive, DFC-15 Multilabel, MLRSNet, and two variants of the BigEarthNet dataset based on two Corine Land Cover (CLC) nomenclatures (Available at https://land.copernicus.eu/user-corner/technical-library/corine-land-cover-nomenclature-guidelines/html (accessed on 4 January 2023 )), CLC with 43 labels (BigEarthNet-43), and CLC with 19 labels (BigEarthNet-19). When analysing the datasets that have hyperspectral RSI (meaning that they have several spectral bands), such as Ankara and BigEarthNet, we only use the RGB spectral band to train the models.

A summary of the properties of the seven datasets is given in Table 1. We can see that the selected datasets are diverse along several lines: number of images, locations, number of labels, image resolution, and number of labels per example image (label cardinality). This means that the trained predictive models are trained and evaluated in a challenging environments. The selection of typical images from the different datasets with their corresponding class labels is given in Figure 1.

*Remote Sens.* **2023**, *15*, 538 

**Table 1.** Description of the used RSI multi-label datasets. $|\mathcal{L}|$ denotes the number of possible labels; *Card* denotes label cardinality (i.e., average number of labels per image); *Dens* denotes label density (average proportion of images labeled with a given label); $N$ is the number of images in the dataset, of which $N_{train}$ are in the train and $N_{test}$ in the test datasets; and $w \times h$ is the dimension of the images (in pixels).

| Dataset | Image Type | $\lvert\mathcal{L}\rvert$ | *Card* | *Dens* | $N$ | $N_{train}$ | $N_{test}$ | $w \times h$ |
|---|---|---|---|---|---|---|---|---|
| Ankara | Hyperspectral/Aerial RGB | 29 | 9.120 | 0.536 | 216 | 171 | 45 | $64 \times 64$ |
| UC Merced Land Use | Aerial RGB | 17 | 3.334 | 0.476 | 2100 | 1667 | 433 | $256 \times 256$ |
| AID Multilabel | Aerial RGB | 17 | 5.152 | 0.468 | 3000 | 2400 | 600 | $600 \times 600$ |
| DFC-15 Multilabel | Aerial RGB | 8 | 2.795 | 0.465 | 3341 | 2672 | 669 | $600 \times 600$ |
| MLRSNet | Aerial RGB | 60 | 5.770 | 0.144 | 109,151 | 87,325 | 21,826 | $256 \times 256$ |
| BigEarthNet | Hyperspectral/Aerial RGB | 19 | 2.900 | 0.263 | 590,326 | 472,245 | 118,081 | $256 \times 256$ |
| BigEarthNet | Hyperspectral/Aerial RGB | 43 | 2.965 | 0.247 | 590,326 | 472,245 | 118,081 | $256 \times 256$ |

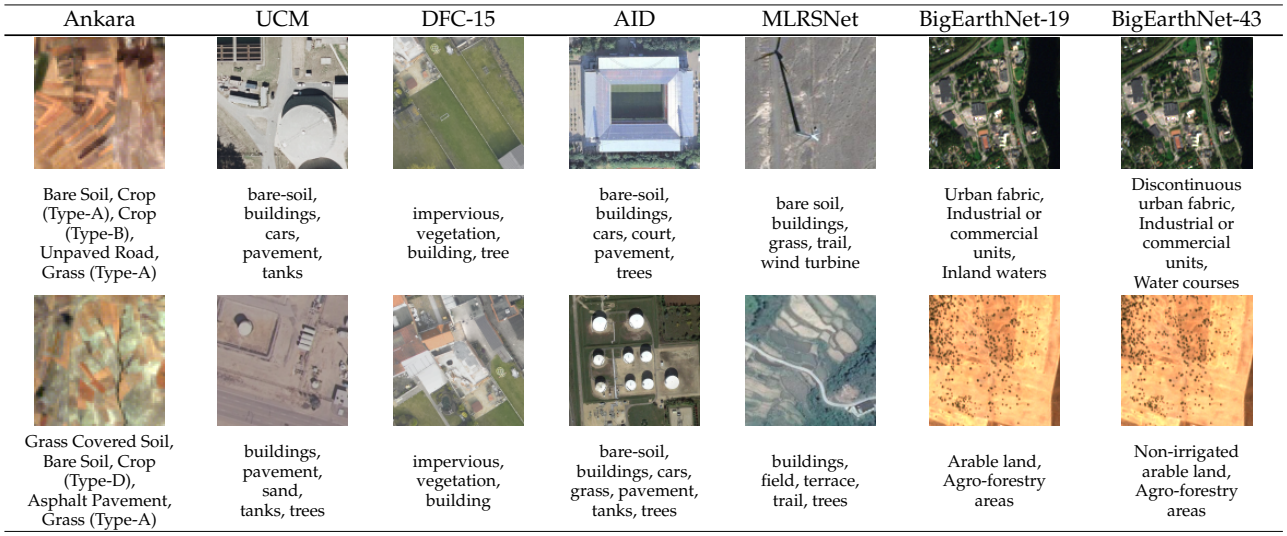



**Figure 1.** An illustration of the diversity of images from the different RSI datasets with the corresponding class labels.

#### 2.1.1. UC Merced Land Use

The UC Merced data set contains 2100 images grouped into 21 broad categories at the scene level. There are a total of 100 images per category, with the size of $256 \times 256$ and a spatial resolution of 0.3 m. The initial version of this dataset was for single-label classification purposes [26]. Later, Chaudhuri et al. [27] relabeled the images with multiple labels. The total number of distinct object-level labels is 17: airplane, bare soil, buildings, chaparral, court, dock, field, grass, mobile home, pavement, sand, sea, ship, tanks, trees, and water. Each image is annotated with one or more (maximum 7) labels at the object level, containing 3.3 object-level labels per image on average.

#### 2.1.2. AID Multilabel

The initial single-label AID dataset [28] was relabeled with multiple labels per image and became the AID multi-label dataset [20]. It is also a more challenging dataset than the U Merced dataset. It contains 3000 aerial images from 30 categories with manually assigned multiple-object labels. The resolution of the images is $600 \times 600$ pixels, where each image has 5.5 object-level labels on average (maximum 11). The spatial resolution varies from 0.3 to 8 m.

#### 2.1.3. Ankara HIS Archive

This is a small hyperspectral dataset containing 216 image tiles with a size of $63 \times 63$ pixels [29]. The image patches are obtained by fragmenting large hyperspectral

images, acquired by the NASA EO-1 satellite's Hyperion sensor from the area surrounding the city of Ankara in Turkey. Each image is associated with multiple object-level labels (land-cover-classes) and a single land-use scene-level label where the ground resolution is 30 m. There are 9 object-level labels per image on average, and a maximum of 17. It contains 119 channels of hyperspectral images and corresponding three-channel (RGB) images. We only use the RGB channels.

#### 2.1.4. DFC-15 Multilabel

The DFC-15 Multilabel dataset [30] is built from a semantic segmentation dataset (DFC15), first used in the 2015 IEEE GRSS data fusion contest. This dataset is acquired over Zeebrugge with an airborne sensor (300 m off the ground). There are a total of 7 tiles, where each of the tiles is 10,000 $\times$ 10,000 pixels with a spatial resolution of 5 cm. The images are assigned pixel-level labels, where each pixel is categorized into 8 object classes: impervious, water, clutter, vegetation, building, tree, boat, and car. The final dataset contains 3342 image patches with 600 $\times$ 600 image resolution, where each image is associated with image-level multi-labels.

#### 2.1.5. MLRSNet

The MLRSNet dataset is an RSI dataset containing optical satellite images with high spatial resolution [31]. It contains 109,161 RSI annotated with 60 predefined class labels, where the number of labels per image varies from 1 to 13. The images have a fixed resolution of 256 $\times$ 256 pixels, where the pixel resolution varies from $\sim$10 m to 0.1 m. This dataset can be used for a wide range of learning tasks, such as multi-label classification, multi-class classification, multi-label image retrieval, and image segmentation. There are 10 images that do not have labels assigned, as shown in Figure 2; therefore, we exclude these images from the experiments. The final dataset contains 109,151 images.

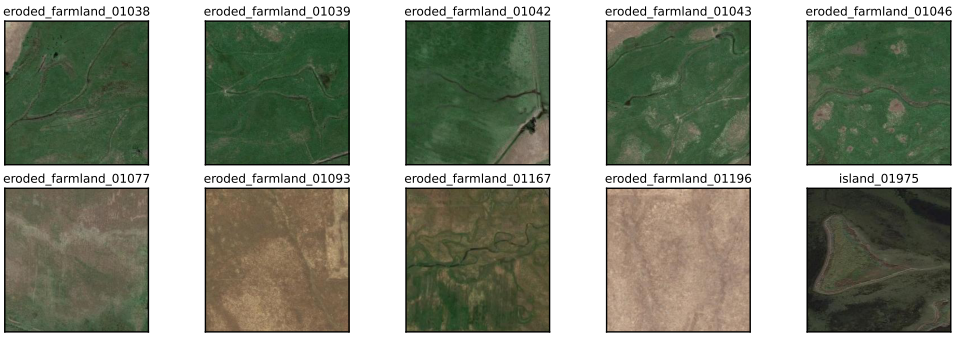



**Figure 2.** Images from the MLRSNet dataset that do not have labels assigned.

#### 2.1.6. The BigEarthNet Archive

BigEarthNet [17] was constructed by the Remote Sensing image Analysis (RSiM) Group and the Database Systems and Information Management (DIMA) Group at the Technische Universität Berlin (TU Berlin). It is the largest dataset of image patches annotated with multiple labels available to date. There are 590,326 such patches for which Sentinel-2/S2 (and later also Sentinel-1/S1) images are available. For S2, twelve channels are available, and for S1, two channels are available. We only use three of the S2 channels (RGB images). Each image patch is annotated by multiple land-cover classes (i.e., multi-labels) taken from the CORINE Land Cover database of the year 2018 (CLC 2018). Originally, 43 labels were used. These were later merged into 19 labels [32].

### *2.2. Overview of the Learning Methods for Multi-Label Classification*

We address the MLC task by applying a Convolutional Neural Network (CNN) as a deep learning approach and exploiting the the transfer learning paradigm. We use the

*Remote Sens.* **2023**, *15*, 538 

weights learned on the ImageNet as the initialization procedure, and we further perform calibration by fine-tuning the models on the training sets of of RSI. The deep models are used in two different scenarios, namely as end-to-end approaches to directly perform the MLC task (Figure 3a,b), and as CNN-based feature extractors (Figure 3a) to generate feature representations, which are used as inputs to the tree-ensemble methods such as Random Forests and Extremely Randomized Trees for MLC (Figure 3c).

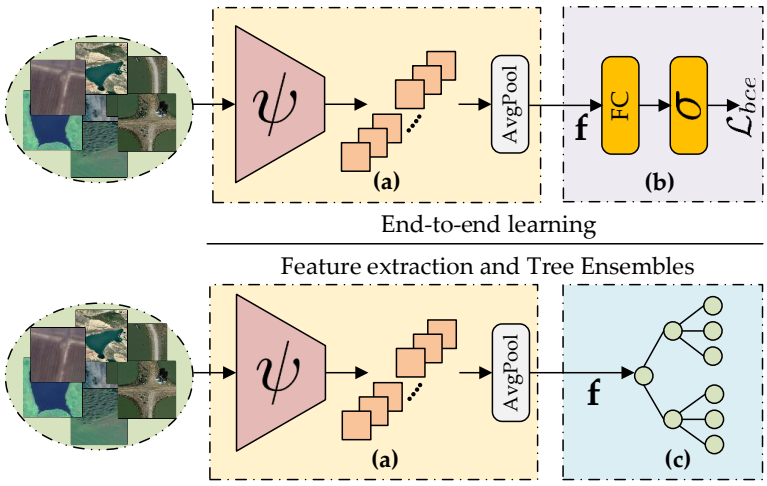



**Figure 3.** Overview of the proposed strategy for MLC of RSI. Part (**a**) of the strategy is the fully convolutional CNN backbone feature extractor. This part is same for both end-to-end learning, and feature extraction. The next part of the strategy, marked (**b**), is only used during end-to-end learning. The last part, marked with (**c**), is used for training tree ensemble methods from the extracted features, as a separate task and not a part of the end-to-end learning process. For this separate task the CNN feature extractor (**a**) is used in offline mode to generate the required feature representations, which serve as inputs to learn tree ensembles.

We make the following modifications to the used network architecture for the MLC task. To make it applicable to differently sized input images, we make sure the backbone CNN is fully convolutional. Next, to reduce the spatial dimension, the feature maps are aggregated through the average pooling layer, where $d$-dimensional feature representations $\mathbf{f}$ are extracted. The classification part is replaced with a single fully connected (FC) layer with the number of output units equal to the number of classes. This can also be seen as a layer providing non-normalized log probabilities $z_i$, i.e., scores (or also called *logit*s), that a classification model generates. This *logit* vector is then converted to a vector of probabilities by using the *sigmoid* activation function $\sigma(z_{i,c}) = 1/(1 + e^{-z_{i,c}})$, where $z_{i,c}$ is the *logit* of the predicted class. The *sigmoid* function is suggested for MLC, instead of the $softmax$ function, as the activation of the last layer of the model, since the probabilities produced by *sigmoid* are independent and are not constrained to sum up to one. It follows a Bernoulli distribution and thus allows multiple label predictions [18]. It is important to mention that this part of the model (Figure 3b) is only used during training in the end-to-end mode to directly perform the MLC task.

The training process uses the standard binary cross-entropy learning objective, which takes the following form:

$$\mathcal{L}_{bce} = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)], \tag{1}$$

where $n$ is the number or samples in a given mini-batch, $y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,c}]$ is a binary vector representing the multi-labels of a given image, and $\hat{y}_i$ is the predicted output vector

of probabilities from the *sigmoid* layer. This learning objective is most commonly used in MLC tasks [18].

#### 2.2.1. Deep Learning Methods

To investigate the impact of the network architecture on the learning process, we consider several popular CNN configurations pre-trained on ImageNet. We use VGG (VGG-16, VGG-19) [23] , ResNet (ResNet-34, ResNet-50, ResNet-152) [24] and EfficientNet (EfficientNet-B0, EfficientNet-B1, EfficientNet-B2) [25] as backbone CNN models for the MLC task. Throughout the experiments, these methods are used either in an end-to-end manner or as CNN-based feature extractors.

- VGGs: VGG is a deep CNN network architecture developed by the Visual Geometry Group (VGG) team [23]. It is also the basis of ground-breaking object recognition models that surpass baselines on many tasks and datasets beyond ImageNet and is still one of the most popular image recognition architectures. Two variants of this family of architectures are intensively studied for their performance—VGG-16 and VGG-19. The VGG-16 model can be seen as an upgrade of AlexNet, while VGG-19 is similar to VGG-16 but contains more layers. They are modeled in such a way that convolutions would actually look simpler by replacing large AlexNet convolution filters with a $3 \times 3$ filter, while padding to maintain the same size before a $2 \times 2$ $MaxPool$ layer down-samples the image size.
- ResNets: ResNets are a family of deep CNN architectures that follow the residual learning principle to ease the training of very deep networks [24]. Their design offers an efficient way to solve the issues related to the vanishing gradients. ResNet follows VGG's full $3 \times 3$ convolutional layer design. The residual block has two $3 \times 3$ convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a Rectified Linear Unit ($ReLU$) activation function. Then, there is a skip (or so-called skip connection) of those two convolution operations, where the input is directly added before the final $ReLU$ activation function. This kind of design requires that the output of the two convolutional layers has to be the same shape as the input, so that they can be added together. By configuring different numbers of channels and residual blocks in the module, we can create different ResNet models, such as the deeper 152-layer ResNets, i.e., ResNet-152. For the experiments, we use three variants of ResNet: ResNet-34, ResNet-50, and ResNet-152.
- EfficientNets: Unlike conventional deep CNNs, which are often over-parameterized, and arbitrarily scale network dimensions, such as width, depth, and resolution, EfficientNets are methods that uniformly scale each dimension with a fixed set of scaling coefficients [25]. These models surpass state-of-the-art accuracy, with up to 10 times better efficiency (i.e., are both smaller and faster than competitors).

#### 2.2.2. Tree Ensemble Methods

In the experiments, we consider two types of ensemble methods based on decision trees for MLC as a main learning model, namely, ensembles based on random forests for MLC [33] and extremely randomized trees for MLC [34], respectively.

- Random Forest: Random forest (RF) is an ensemble learning method for classification and regression, which creates a set of individual decision trees that operate as an ensemble. It uses bagging and feature randomness to create diversity among the predictors: At each node in the decision tree, a random subset of attributes is taken, and the best split is selected from this subset of attributes. Each individual tree in the random forest provides a class prediction, where the predictions can be aggregated by taking the average (for regression tasks) and the majority or probability distribution vote (for classification tasks). RFs were adapted for the task of MLC [33].
- Extremely Randomized Trees: Extremely Randomized Trees, or so-called Extra Trees (ET), is also an ensemble learning method similar to the Random Forest, which is based on extreme randomization of the tree construction algorithm. As compared

to the Random Forest ensemble, it operates with two key differences: it splits nodes by choosing the cut-points fully at random, and it uses the whole learning sample to grow the trees. The randomness in this method comes from the random splits of all observations, rather then bootstrapping the data as in RF. ETs were adapted for the task of MLC [34].

## 3. Experimental Design

This section presents the details of the experimental study design. It includes a detailed overview of the experimental setup describing the different learning settings in the end-to-end approaches and the tree ensemble methods. Next, we describe the evaluation strategy of the partitioning of the image datasets into disjoint splits used for training and testing MLC models. Finally, we describe the evaluation measures used to assess the predictive performance of the different methods as well as the statistical procedures used to analyze the results.

The experimental study is tailored to answer the following research questions:

(i) What is the influence of the learning strategy on the performance of end-to-end approaches: Is fine-tuning or pre-training only more suitable for solving the RSI-MLC task?
(ii) Which network architecture is the best choice for end-to-end MLC of RSI and for use as a feature extractor and further training of tree ensembles for MLC?
(iii) How do end-to-end learning and feature extraction plus tree ensembles compare on the task of RSI for MLC (assessed by using the best performing architecture from the previous analysis)? and
(iv) How does the number of training examples influence the predictive performance of the methods used?

### *3.1. Experimental Setup*

#### 3.1.1. End-to-End Learning Approaches

We train the deep network models by using two different learning strategies based on transfer learning. Both settings use the *ImageNet weights* for initialization, while fine-tuning different parts of the backbone CNN model. In the first learning setting, shown in Figure 4a, we fix all the model layers pre-trained on ImageNet and directly *t*ransfer the learned knowledge to the target domain. In the second learning setting, shown in Figure 4b, the entire network architecture is being fine-tuned to the new target domain. In both settings, we use a single application-dependent fully connected layer learned from scratch (i.e., the classification layer).

We use the binary cross entropy as an objective function (see Equation (1)) to optimize the model parameters, and apply the same training procedure and hyperparameters for 100 epochs across all datasets. The optimization of the model parameters is performed with the Adam optimizer with a learning rate of $1 \times 10^{-4}$ and a mini-batch of 64.

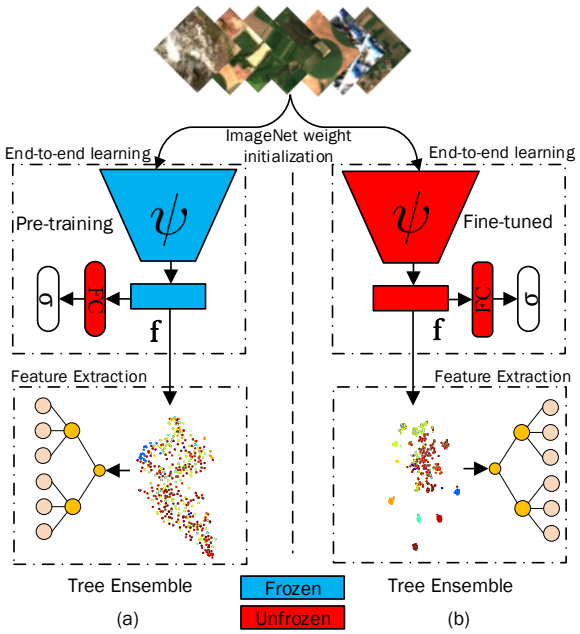



**Figure 4.** Two distinctive learning strategies for the MLC task. In the first learning setting, marked as (**a**), we rely on the ImageNet pre-training, where all the model layers are frozen, except the last fully connected layer, which is trained for the new target domain. In the second learning setting, marked as (**b**), the entire network architecture is fine-tuned along with the newly added fully connected layer. Moreover, we use both approaches either as feature extractors (in "offline" mode) jointly with tree ensemble methods, or as end-to-end learning machines to directly perform the MLC task. Note that the parts of the model highlighted with blue indicate that there is no update of the parameters in the model during the training process; on the other hand, with red, we highlight the opposite task, which means the parameters of the model are unfrozen and updated during training.

To prevent overfitting, we use data augmentation and modify data on the fly, so that our CNN models are transformation-invariant to the maximum possible extent. Moreover, we use data augmentation because we want to ensure that the predictive performance and generalization capability of the learned model is preserved to some extent, especially when training a predictive model on a dataset that contains a limited number of images (e.g., the Ankara dataset). Using the `Albumentations` library [35] (Available at: https://albumentations.ai (accessed on 4 January 2023 )), we performed the following image transformations: horizontal flipping, random rotations in the range $\pm 10\%$, scaling by a factor in the range (0, 0.15), shifting by a factor in the range (0, 0.1), random crops with 50% of the original image size, random brightness within the range of $(-0.3, 0.3)$, and random contrast within the range of $(-0.3, 0.3)$. In addition, we consider the application of the following transformations to the input image: Contrast Limited Adaptive Histogram Equalization (CLAHE) to the input image, where the size of the grid for histogram equalization is set to $8 \times 8$ pixels; blurring with Gaussian kernel with $\sigma$ value in the range (3, 7); median blur with aperture linear size value $v$ randomly sampled once per image in the range (3, 7); motion blur with kernel size $\omega$ for blurring the input image, randomly chosen once per image in the range (3, 7); or Gaussian noise, where the variance range for noise is taken from the interval (10, 50).

The augmentations are performed in random order and with a 50% chance, which means that in such a setting there might be no augmentations applied over an image at all. This is important because, during the training process, the model needs to see the original image at least once in order to make reasonable predictions afterwards [36]. Some example augmentations are shown in Figure 5. Each row represents a specific dataset, where the left-most image marked with red is the original version, while the remaining images are the

*Remote Sens.* **2023**, *15*, 538 

augmented versions of the same image. Note that multiple augmentations could be applied on the same image. Because of that, we are not mixing certain image augmentations, such as the blurring operations (i.e., Gaussian, median and motion blur) with color (i.e., CLAHE) and Gaussian noise corruption transformations, simultaneously. The main reason for this is that we want to avoid strong image degradation and loss of contextual information, which is crucial in remote sensing imagery.

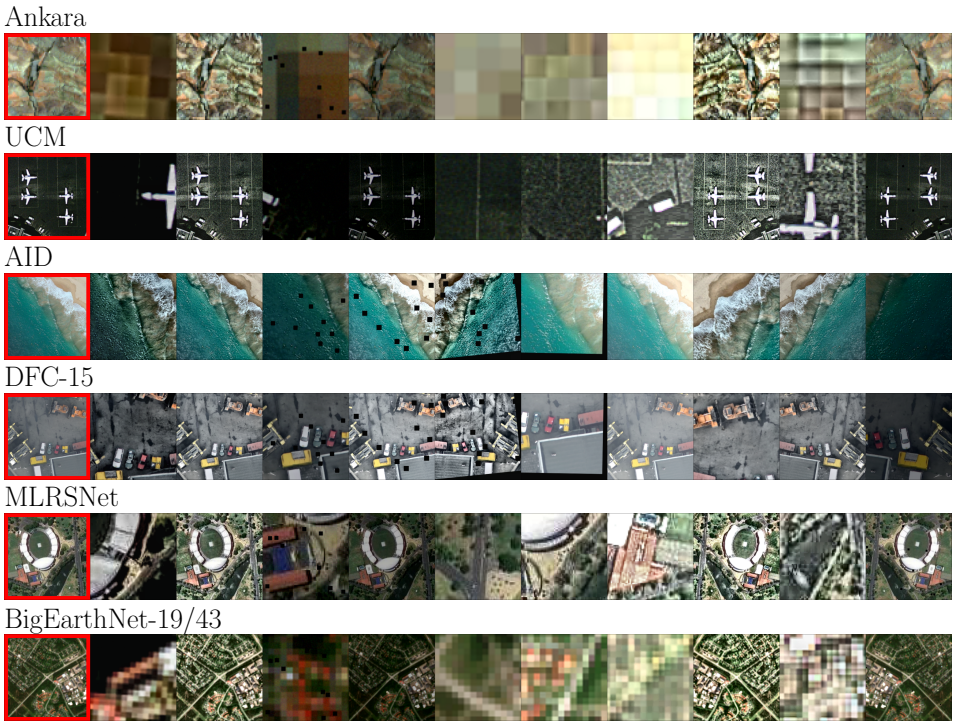



**Figure 5.** Augmentation examples. The left most image in each row marked with red is the original version, while the remaining images are the augmented versions. We can see the extent of variability added to the training datasets by the augmented images.

#### 3.1.2. CNNs as Feature Extractors and Tree Ensembles

The deep learning methods presented in Section 2.2.1, are further used as feature extractors, for each of the two learning settings described in Section 3.1.1. Hence, we rely on two feature representations $\mathbf{f} \in \mathbb{R}^d$ extracted from the CNN feature extractors: representations based on pre-training (Figure 4a), and fine-tuning (Figure 4b). Furthermore, each feature representation is used as input to the tree ensemble methods, i.e., Random Forests (RF) and Extremely Randomized Trees (ERT)/Extra Trees (ET) for short. We use 150 base models in each of the ensembles and `sqrt` as the feature subset size.

### *3.2. Evaluation Strategy*

We conducted experiments over the several publicly available RSI datasets described above and explored how the performance of the learning strategies is affected by data availability, image quality, and label dimensionality. To tackle this problem, we included datasets which are less challenging in terms of image resolution, such as DFC-15 and MLRSNet, and datasets such as Ankara, which is very limited in size and has poor image quality. To evaluate the effectiveness of the methods, we perform the splitting according to the datasets that already contain a predefined subset of images, such as AID and DFC-15, where the train and test ratios are approximately 80% and 20%. The remaining dataset is partitioned accordingly, by adopting an iterative stratified sampling strategy in order to preserve the relative frequency of the labels in the datasets to the maximum possible extent.

Moreover, we split a validation set of 10% from the overall training data, which is only used to monitor the learning progress and to provide an unbiased estimate of the model fit when tuning hyperparameters. A description of the datasets after the splitting procedure is given in Table 1.

### *3.3. Evaluation Measures and Statistical Analysis*

Many evaluation measures are used to assess the predictive performance and effectiveness of MLC methods, offering different viewpoints on the performance of the methods. The evaluation measures can be grouped into two groups: measures based on bipartitions (example-based and label-based measures) and ranking-based evaluation measures [8]. The example-based evaluation measures compute the average difference between the true labels and the predicted labels for each data point, averaged across all the examples in the dataset. Unlike example-based measures, label-based measures evaluate each label separately and then average the performances across all labels. The ranking-based evaluation measures compare the predicted ranking of the labels with the ground truth ranking (where all present labels are ranked before all absent labels).

In this study, we present the results in terms of ranking-loss as an evaluation measure. Ranking loss evaluates the average fraction of label pairs that are misordered for a given example. Note, however, that we provide complete results in terms of all other evaluation measures for the MLC task in the Appendix A section (for the calculation of the evaluation measures, we use the `sckit-learn` implementation [37]). We focus on ranking loss, since we believe it is one of the best indicators for measuring the performance of methods for MLC. Moreover, this measure is threshold-independent, which means we do not rely on the use of techniques for threshold estimation to produce the predicted labels. Ranking loss is defined as follows:

$$rl = \frac{1}{N}\sum_{i=1}^{N}\sum_{(j,k):y_j>y_k}\left(I[r_i(j) < r_i(k)] + \frac{1}{2}I[r_i(j) = r_i(k)]\right), \quad (2)$$

where $y_i$ and $\hat{y}_i$ are the true and the predicted labels, respectively, $N$ is the number of examples, $r_i(j)$ is the ranking of label $j$ for instance $x_i$, and $I$ is an indicator function. The smaller the value of $rl$, the better the performance.

We use the Friedman test to assess whether the overall differences in performance of the used approaches evaluated across the RSI datasets are statistically significant and the post hoc Nemenyi test to detect between which methods the statistically significant differences occur. The obtained results are presented in the form of Nemenyi post hoc average rank diagrams [38] for the ranking loss measure. In the analysis, the significance level was set to $\alpha = 0.05$. The best-performing methods are on the left-most side of the diagram along the axis (average ranks closer to 1), and the methods whose predictive performance does not differ significantly at $\alpha = 0.05$, are connected with a red line.

### *3.4. Implementation Details*

We implemented the deep learning models in the `PyTorch` framework (Available at: https://pytorch.org, accessed on 4 January 2023 ). We use the built-in implementations of VGGs, ResNets and EfficientNets, initialized with weights learned on ImageNet. We modify each of the backbone models in order to accept arbitrary-sized input images by replacing the last max pooling layer with the average pooling operation with a kernel size of 1, resulting in the following $d$-dimensional feature representations: $d = 4096$ for VGG-16 and VGG-19, $d = 512$ for ResNet-34, and $d = 2048$ for ResNet-50 and ResNet-152. We have 1280 output dimensions for EfficientNet-B0 and EfficientNet-B1, while EfficientNet-B2 has 1408 dimensions. Finally, for the tree ensemble methods, we use the `scikit-learn` [37] implementation of Random Forest tree ensembles for multi-label classification (MLC) [33] and Extra Tree ensembles for MLC [34]. The tree ensembles for MLC simultaneously predict the probability of each of the multiple class labels. The labels can then be ranked based on

these predicted probabilities. The complete source code and the datasets used to execute the study are publicly and freely available at https://github.com/marjanstoimchev/RSMLC (accessed on 4 January 2023).

## 4. Results and Discussion

This section presents the results of our experimental study in MLC of RSI. It answers one of the experimental questions posed in Section 3 in each subsection. It thus discusses (1) the influence of the learning strategy, (2) the comparison of different network architectures used as end-to-end approaches, as well as feature extractors, (3) the comparison between end-to-end methods and tree ensembles, and (4) the influence of the number of available labeled images on predictive performance.

### *4.1. The Influence of the Learning Strategy*

In the first experiment, we explored whether fine-tuning or pre-training only is the more appropriate learning strategy (as defined in Section 3.1.1). To provide the answer to this question, we conducted experiments by training different network architectures over the datasets, and using them in two different modes of operation: (1) as feature extractors providing the feature representations to the tree ensembles, and (2) as end-to-end learning methods. Moreover, for each learning approach, we present the difference in performance in the form of a heat map, which can be formally defined as follows:

$$\mathbf{H} = \begin{bmatrix} rl_{11} & rl_{12} & \cdots & rl_{1n} \\ rl_{21} & rl_{22} & \cdots & rl_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ rl_{m1} & rl_{m2} & \cdots & rl_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}, \tag{3}$$

where $rl_{i,j} = rl_{i,j}^{fine-tune} - rl_{i,j}^{pre-train}$ is the difference between fine-tuning ($rl^{fine-tune}$) and pre-training ($rl^{pre-train}$) in terms of the ranking loss measure, calculated for the $i$-th dataset and the $j$-th method, respectively. This is performed for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, where $m$ denotes the number of datasets and $n$ is the number of methods/CNN architectures.

These differences in performance are presented in Figure 6. We observe that for smaller datasets (e.g., Ankara, UCM, etc.), pre-training is the preferred choice, which can be mostly seen for the VGG architectures trained in an end-to-end manner over the DFC-15 dataset, where the largest differences in ranking loss are observed. These differences are slightly smaller for the DFC-15 dataset and the tree-ensemble methods. On the other hand, the fine-tuned versions of the models perform significantly better when data availability is not a problem. Furthermore, for the tree ensemble methods, the differences in performance are significantly increased (and more on the positive side), which points to the fact that the quality of the feature representations is of great importance for the tree ensemble methods to further boost their predictive performance. Overall, we can conclude that by solely relying on the ImageNet pre-training, we end up with worse model performance in almost all cases, because the content present in RSI is quite complex as compared to images present in the ImageNet dataset. More detailed results of the analysis are presented in Table A1 (Appendix A), where the performance figures for the ranking loss measure are given.

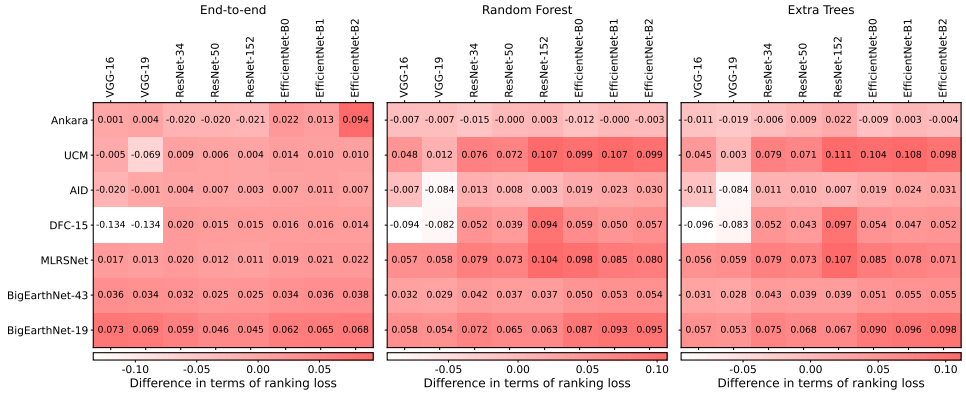
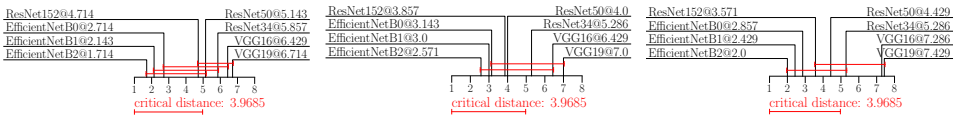



**Figure 6.** Comparison between fine-tuning and pre-training in terms of ranking loss. The results for the different architectures are presented in the form of a heat map showing the difference in performance between the learning approaches. Negative values indicate that pre-training is better than fine-tuning and positive when fine-tuning is better than pre-training. The intensities of the colors in the heat map are directly related to the difference in performance between the learning approaches (CNN architectures). The datasets on the y-axis are ordered by the number of images they contain.

### *4.2. Comparison of Different Network Architectures*

Based on the analysis from Section 4.1, we used the fine-tuning approach to further explore which network architecture is the most suitable choice for RSI MLC tasks. The results are shown in terms of ranking loss and in the form of average rank diagrams. From Figure 7, we see that the EfficientNet variants, especially EfficientNet-B2, tend to produce the better results as compared to other network architectures. The differences in performance are statistically significant as compared to the VGG variants. The results also reveal that the EfficientNet-B2 model, used as base feature extractor in combination with tree ensemble methods, such as RF as in Figure 7b and ET in Figure 7c, is the clear winner, which indicates that this network architecture is the best choice for this task. Moreover, the ResNet-based network architectures are the closest competitors to EfficientNet variants (both when used with tree ensembles and in an end-to-end manner), i.e., the ResNet-152 model. Although the EfficientNet variants are not statistically significantly better than ResNet-152, they are by far more lightweight in terms of model parameters (e.g., EfficientNet-B0 is approximately 11× smaller, EfficientNet-B1 is 8×, and EfficientNet-B2 is approximately 6.5× smaller than ResNet-152, respectively). Comparison between different network architectures in terms of other MLC performance measures in the form of average rank diagrams is given in Figures A1–A6 in the Appendix A (Figures A1–A3 for fine-tuned features and label-based, example-based and ranking-based measures, respectively, Figures A4–A6 for pre-trained features and label-based, example-based and ranking-based measures, respectively).




(**a**) End-to-end (**b**) Random forest (**c**) Extra trees

**Figure 7.** Comparison between different network architectures in terms of ranking loss. The results are presented in the form of average rank diagrams at 0.05 significance level for (**a**) End-to-end learning, (**b**) Random forests and (**c**) Extra trees. The best ranking methods are at the left-most side of the diagram. The difference in performance among the methods connected with a red line is not statistically significant.

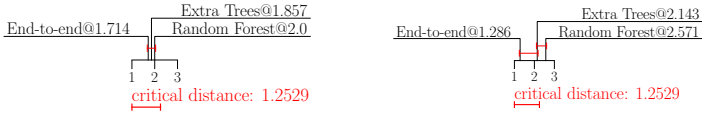
Overall, we can conclude that in the fine-tuning setting, it is favorable to use lightweight network architectures in terms of model parameters for end-to-end learning, which are

*Remote Sens.* **2023**, *15*, 538 

also capable of learning more discriminative feature representations when used as feature extractors. This is in direct relation to their capability of capturing high-level content present in RSI to a greater extent as compared to the other network architectures. Moreover, they are the preferred choice when addressing the problems encountered in challenging deployment scenarios, which means they can maintain good predictive performance while keeping the computational costs at a reasonable level.

### *4.3. Comparison of Different Learning Approaches*

To answer the experimental question of how end-to-end learning and feature extraction plus the tree ensembles compare in the task of RSI MLC, we present the results in the form of average rank diagrams, where we use the pre-trained and fine-tuned versions of the EfficientNet-B2 model. Recall that EfficientNet-B2 produced the best results overall, either when used as a feature extractor where the extracted feature representations are further utilized in the tree ensembles, or as an end-to-end approach to directly address the MLC task. The results are shown in Figure 8a for fine-tuning and in Figure 8b for pre-training only.




(**a**) Fine-tuning (**b**) Pre-training

**Figure 8.** Comparison between different learning methods in terms of ranking loss. The results are presented in the form of average rank diagrams at 0.05 significance level for (**a**) Fine-tuning and (**b**) Pre-training only. The best ranking methods are at the left-most side of the diagram. The differences among the methods connected with a red line is not statistically significant.

As seen from the diagrams, end-to-end learning is ranked best, followed by Extra trees and Random Forests. In the case of fine-tuning (Figure 8a), there is no statistically significant difference among the classification methods. In the pre-training setting, the end-to-end learning approach is significantly better than the random forest method. Overall, the differences in predictive performance depend on the specific deep neural network architecture and the learning setting. Comparison between different learning methods in terms of other MLC performance measures in the form of average rank diagrams is given in Figures A7 and A8 in the Appendix A. Figure A7 concerns the use of fine-tuned and Figure A8 the use of pre-trained features.

### *4.4. Influence of the Number of Available Labeled Images*

To answer the fourth question about the relation of the number of available images and the performance of the different MLC models, we conducted additional experiments focusing on the biggest dataset available—BigEarthNet. We design an experimental protocol which partitions the BigEarthNet dataset into distinctive subsets of images, namely, fractions of: 0.1%, 0.5%, 1%, 5%, 10%, 25%, and 50% of 590,326 images, respectively. Each of the fractions is further split into disjoint subsets of training, validation, and testing images, with sizes 70%, 10%, and 20% of the fraction size, respectively. Moreover, the test sets are built in a cumulative manner, which means the test subset of images from the previous fractions are inherited into the test set of the next fraction. By doing this, we are evaluating the effectiveness of the models on new subsets of images, while taking the old ones into account, i.e., we simulate a scenario where we add new images. The sampling of the fractions and the subsets within the fractions can be done in two different ways: (i) sampling with stratification, and (ii) random sampling. Furthermore, we are also assessing the generalization capability of the model when exposed to different distribution shifts. To obtain a more reliable estimate of the predictive performance, we repeated the experiment five times and calculated the average performance and standard deviation ($\mu \pm \sigma$) in terms of ranking loss.

To carry out the experiment, we used the two versions of the BigEarthNet dataset, namely, BigEarthNet with 19 and 43 CLC nomenclatures. We selected this dataset because it is the largest one in terms of the number of images. For these experiments, we used EfficientNet-B2 as network architecture, since it produced the best results overall in the previous experiments. We fine-tuned the model parameters for 25 epochs and used the trained model as a feature extractor for the tree ensembles, as well as an end-to-end approach. We present the results in the form of learning curves in order to see how the number of training examples influences the predictive performance. The results of the experiment for the BigEarthNet-19 and BigEarthNet-43 are shown in Figure 9. We can see that the performance of the learning methods in all cases improves of the total number of training examples by up to 10%, after which it degrades. End-to-end learning methods perform worse than tree ensemble methods in all cases in the learning curve, but the differences in performance are only visible in the case of stratified sampling. The differences are more expressed in the BigEarthNet-19 dataset (Figure 9). This means that the different learning methods are affected differently by the choice of the sampling strategy.

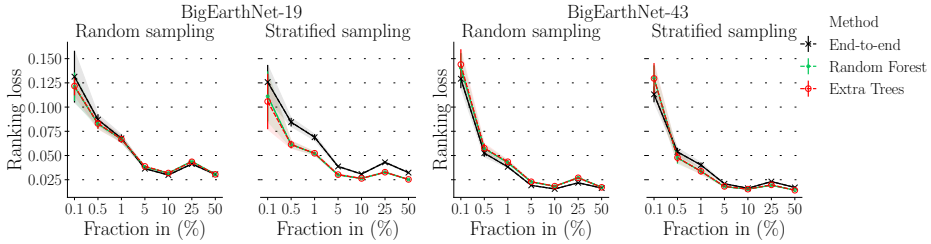



**Figure 9.** Comparison between different learning methods in terms of ranking loss when evaluated on different fractions of labeled examples from the BigEarthNet-19 and BigEarthNet-43 datasets,. The results are shown as learning curves depicting $\mu \pm \sigma$ for ranking loss across five repeats. The y-axis is shared across the figures.

## 5. Conclusions

In this study, we have presented a comparative analysis of methods for multi-label classification of remote sensing imagery. We have compared several popular deep learning methods based on two modes of operation, where they are (1) used as feature extractors in a combination with tree ensemble methods such as random forests and extra trees, and (2) used as end-to-end approaches to directly address the MLC task.

We focused on four aspects: (1) different transfer learning paradigms, namely, learning features based on ImageNet pre-training only, as well as learning features with fine-tuning, where the whole network architecture and the model parameters are trained on the new target domain of interest; (2) comparison between different network architectures; (3) comparison between tree ensembles and end-to-end approaches; and (4) investigating the influence of the number of labeled examples on the relative predictive performance of MLC methods. In the first dimension, we showed that it is beneficial to fine-tune the models on RSI to improve their performance. However, in certain cases, where the number of data is limited, it is better to extract features with ImageNet pre-training and only fine-tune the last layer for classification. Furthermore, we showed that it is very important to choose a proper network architecture: EfficientNets proved to be overall the most suitable choice for the task of MLC. They have significantly fewer parameters as compared to the ResNet variants, and no statistically significant difference in predictive performance is observed. We also showed that having the right feature extractor plays an important role in boosting the performance of tree ensemble methods, so that they outperform the end-to-end approaches in certain cases. In the last dimension, we investigated the influence of the amount of labeled data from the BigEarthNet dataset on the relative performance of MLC methods, where we applied two types of sampling strategies: random sampling and sampling with stratification. We showed that in such a setting, the tree ensemble methods

outperform the end-to-end approaches, with the difference in performance clearly visible for stratified sampling.

Considering the findings from this study, further extension of this work should focus on several aspects. Firstly, we will incorporate even a wider range of deep learning models specially devised for the RS MLC task. Next, we will take into account different MLC loss functions to analyze whether they are more suitable for the RS MLC task. Moreover, since the label space of BigEarthNet dataset is organized in a hierarchical manner, we will further analyze the effect of using the hierarchical information in the tree ensemble methods and in the end-to-end approaches. Lastly, we will focus on the application of the methods in a semi-supervised learning setting, where we will exploit the abundance of the unlabeled data in RS domain and investigate if the semi-supervised counterparts can surpass the performance of the supervised learning methods.

**Author Contributions:** Conceptualization, S.D.; methodology, M.S., D.K., and S.D.; software, M.S.; validation, M.S. and D.K.; writing—original draft preparation, M.S.; writing—review and editing, D.K. and S.D.; supervision, S.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** We acknowledge the support of the European Space Agency ESA through the activity of the AiTLAS - AI4EO rapid prototyping environment. This work was also partially supported by the Slovenian research agency through the knowledge technologies program P2-0103 and the project J2-2505.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The AID Dataset in this study is openly and freely available at https://github.com/Hua-YS/AID-Multilabel-Dataset (accessed on 1 July 2022). The DFC-15 dataset in this study is openly and freely available at https://drive.google.com/drive/folders/1TKGS6TIRxQ6a7gdaj0cHs-mRCtv_J1HA (accessed on 1 July 2022). The MLRSNet dataset in this study is openly and freely available at https://data.mendeley.com/datasets/7j9bv9vwsx/2 (accessed on 1 July 2022). UCM dataset in this study is openly and freely available at https://drive.google.com/file/d/1DtKiauowCB0ykjFe8v0OVvT76rEfOk0v/view (accessed on 1 July 2022). The Ankara dataset in this study is openly and freely available at https://bigearth.eu/datasets (accessed on 1 July 2022). The BigEarthNet-19 and BigEarthNet-43 in this study are openly and freely available at https://bigearth.net/ (accessed on 1 July 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

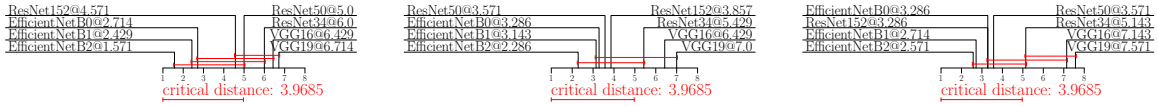
The following abbreviations are used in this manuscript:

| | |
|---|---|
| RSI | Remote Sensing Images |
| MLC | Multi-Label-Classification |
| DNN | Deep Neural Network |
| CNN | Convolutional Neural Network |
| DL | Deep Learning |
| RNN | Recurrent Neural Network |
| LSTM | Long Short-Term Memory |
| GCN | Graph Convolutional Network |
| CLC | Corine Land Cover |
| ReLU | Rectified Linear Unit |
| RF | Random Forest |
| ET | Extra Trees |

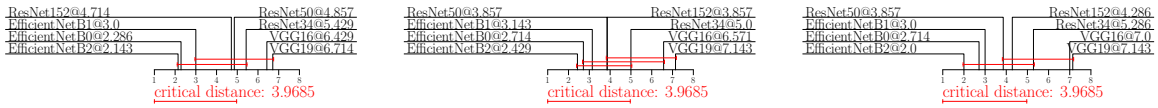
## Appendix A. Complete Results from the Experimental Evaluation

**Table A1.** The performance in terms of ranking loss measure of fine-tuning and pre-training feature learning approaches with different network architectures and different MLC approaches. The best performing network architecture for each dataset is highlighted with green.

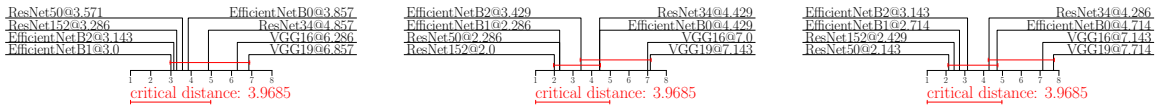
| | | VGG-16 | VGG-19 | ResNet-34 | ResNet-50 | ResNet-152 | EfficientNet-B0 | EfficientNet-B1 | EfficientNet-B2 |
|---|---|---|---|---|---|---|---|---|---|
| **Approach** | **Datasets** | | | | **Pre-Training** | | | | |
| End-to-end | Ankara | 0.298 | 0.371 | 0.343 | 0.351 | 0.350 | 0.349 | 0.330 | 0.422 |
| | UCM | 0.186 | 0.180 | 0.154 | 0.149 | 0.135 | 0.194 | 0.185 | 0.184 |
| | AID | 0.215 | 0.208 | 0.171 | 0.181 | 0.179 | 0.198 | 0.194 | 0.188 |
| | DFC-15 | 0.176 | 0.176 | 0.147 | 0.134 | 0.134 | 0.127 | 0.120 | 0.113 |
| | MLRSNet | 0.347 | 0.360 | 0.306 | 0.240 | 0.229 | 0.318 | 0.300 | 0.326 |
| | BigEarthNet-19 | 0.557 | 0.550 | 0.461 | 0.399 | 0.391 | 0.460 | 0.476 | 0.478 |
| | BigEarthNet-43 | 0.578 | 0.546 | 0.480 | 0.431 | 0.410 | 0.468 | 0.481 | 0.480 |
| Random Forest | Ankara | 0.322 | 0.320 | 0.324 | 0.329 | 0.388 | 0.317 | 0.337 | 0.345 |
| | UCM | 0.381 | 0.398 | 0.469 | 0.420 | 0.539 | 0.495 | 0.508 | 0.468 |
| | AID | 0.250 | 0.244 | 0.265 | 0.247 | 0.294 | 0.268 | 0.257 | 0.262 |
| | DFC-15 | 0.297 | 0.337 | 0.235 | 0.201 | 0.342 | 0.259 | 0.222 | 0.242 |
| | MLRSNet | 0.529 | 0.545 | 0.566 | 0.549 | 0.615 | 0.587 | 0.567 | 0.548 |
| | BigEarthNet-19 | 0.534 | 0.525 | 0.588 | 0.543 | 0.532 | 0.637 | 0.670 | 0.669 |
| | BigEarthNet-43 | 0.547 | 0.537 | 0.600 | 0.557 | 0.545 | 0.654 | 0.686 | 0.683 |
| Extra Trees | Ankara | 0.318 | 0.312 | 0.331 | 0.325 | 0.370 | 0.328 | 0.320 | 0.330 |
| | UCM | 0.390 | 0.405 | 0.457 | 0.417 | 0.552 | 0.483 | 0.513 | 0.462 |
| | AID | 0.254 | 0.250 | 0.255 | 0.253 | 0.305 | 0.265 | 0.250 | 0.256 |
| | DFC-15 | 0.297 | 0.338 | 0.235 | 0.204 | 0.351 | 0.235 | 0.218 | 0.229 |
| | MLRSNet | 0.530 | 0.545 | 0.567 | 0.549 | 0.620 | 0.573 | 0.556 | 0.534 |
| | BigEarthNet-19 | 0.539 | 0.528 | 0.608 | 0.567 | 0.556 | 0.658 | 0.693 | 0.698 |
| | BigEarthNet-43 | 0.550 | 0.540 | 0.622 | 0.581 | 0.570 | 0.676 | 0.709 | 0.713 |
| | | | | | Fine-tuning | | | | |
| End-to-end | Ankara | 0.294 | **0.285** | 0.377 | 0.356 | 0.360 | 0.335 | 0.353 | 0.322 |
| | UCM | 0.224 | 0.508 | 0.101 | 0.097 | 0.112 | 0.088 | 0.096 | **0.081** |
| | AID | 0.265 | 0.202 | 0.152 | 0.143 | 0.147 | 0.137 | **0.131** | 0.137 |
| | DFC-15 | 0.433 | 0.433 | 0.068 | 0.075 | 0.067 | 0.054 | 0.046 | 0.050 |
| | MLRSNet | 0.180 | 0.223 | 0.093 | 0.091 | 0.088 | **0.082** | 0.084 | 0.084 |
| | BigEarthNet-19 | 0.276 | 0.282 | 0.235 | 0.236 | 0.210 | 0.207 | 0.203 | **0.202** |
| | BigEarthNet-43 | 0.271 | 0.276 | 0.243 | 0.232 | 0.199 | 0.206 | 0.195 | **0.194** |
| Random Forest | Ankara | 0.318 | 0.319 | 0.344 | 0.338 | 0.345 | 0.304 | 0.335 | 0.320 |
| | UCM | 0.182 | 0.323 | 0.103 | 0.098 | 0.103 | 0.106 | 0.104 | 0.106 |
| | AID | 0.245 | 0.197 | 0.146 | 0.144 | 0.149 | 0.138 | 0.138 | 0.137 |
| | DFC-15 | 0.433 | 0.433 | 0.050 | 0.047 | 0.050 | 0.046 | **0.041** | 0.044 |
| | MLRSNet | 0.185 | 0.221 | 0.104 | 0.103 | 0.102 | 0.093 | 0.095 | 0.090 |
| | BigEarthNet-19 | 0.258 | 0.268 | 0.229 | 0.219 | 0.214 | 0.222 | 0.219 | 0.217 |
| | BigEarthNet-43 | 0.255 | 0.267 | 0.235 | 0.230 | 0.219 | 0.228 | 0.221 | 0.224 |
| Extra Trees | Ankara | 0.364 | 0.348 | 0.342 | 0.322 | 0.362 | 0.313 | 0.343 | 0.321 |
| | UCM | 0.177 | 0.334 | 0.103 | 0.102 | 0.102 | 0.100 | 0.098 | 0.097 |
| | AID | 0.248 | 0.194 | 0.144 | 0.146 | 0.146 | 0.135 | 0.134 | 0.136 |
| | DFC-15 | 0.433 | 0.433 | 0.049 | 0.050 | 0.046 | 0.046 | 0.041 | 0.043 |
| | MLRSNet | 0.184 | 0.221 | 0.106 | 0.104 | 0.103 | 0.097 | 0.100 | 0.094 |
| | BigEarthNet-19 | 0.257 | 0.268 | 0.227 | 0.217 | 0.213 | 0.222 | 0.218 | 0.216 |
| | BigEarthNet-43 | 0.255 | 0.267 | 0.233 | 0.228 | 0.217 | 0.227 | 0.219 | 0.224 |



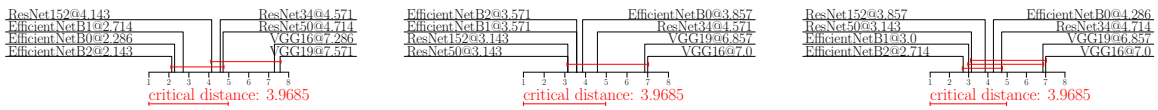

(**a**) End-to-end: Micro-F1 (**b**) Random forest: Micro-F1 (**c**) Extra trees: Micro-F1



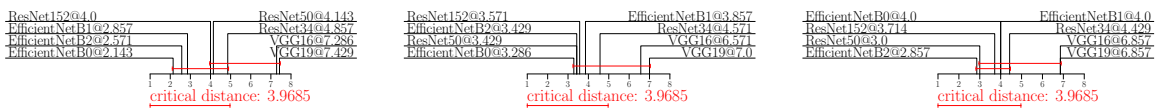

(**d**) End-to-end: Micro-recall (**e**) Random forest: Micro-recall (**f**) Extra trees: Micro-recall



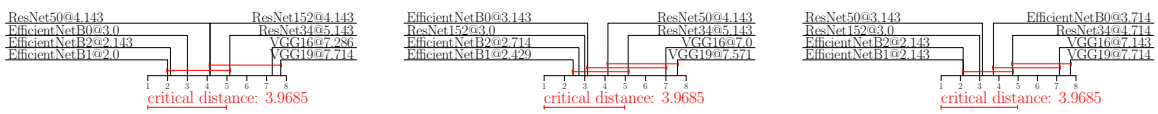

(**g**) End-to-end: Micro-precision (**h**) Random forest: Micro-precision (**i**) Extra trees: Micro-precision




(**j**) End-to-end: Macro-F1 (**k**) Random forest: Macro-F1 (**l**) Extra trees: Macro-F1

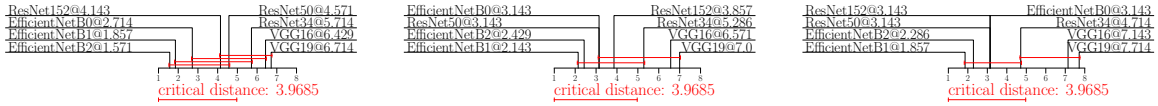



(**m**) End-to-end: Macro-recall (**n**) Random forest: Macro-recall (**o**) Extra trees: Macro-recall

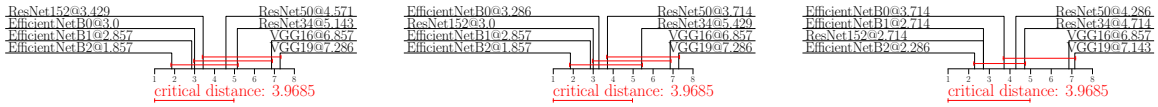



(**p**) End-to-end: Macro-precision (**q**) Random forest: Macro-precision (**r**) Extra trees: Macro-precision

**Figure A1.** Performance of different network architectures in terms of label-based evaluation measures for fine-tuned features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

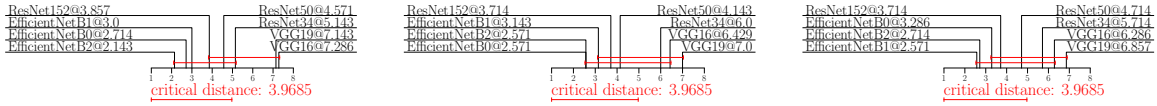



(**a**) End-to-end: Hamming loss (**b**) Random forest: Hamming loss (**c**) Extra trees: Hamming loss

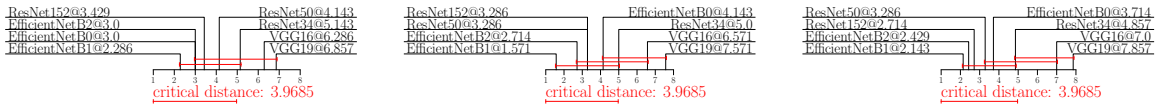



(**d**) End-to-end: Subset Accuracy (**e**) Random forest: Subset Accuracy (**f**) Extra trees: Subset Accuracy

**Figure A2.** Performance of different network architectures in terms of example-based evaluation measures for fine-tuned features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

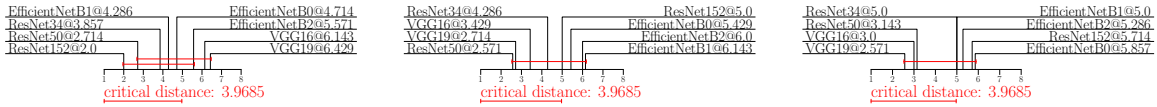



(**a**) End-to-end: Coverage (**b**) Random forest: Coverage (**c**) Extra trees: Coverage

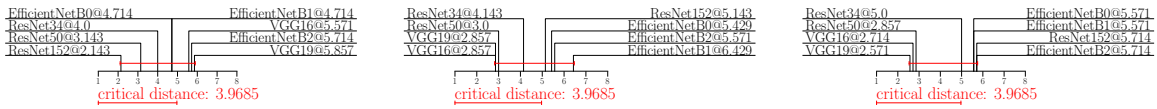



(**d**) End-to-end: Average Precision (**e**) Random forest: Average Precision (**f**) Extra trees: Average Precision

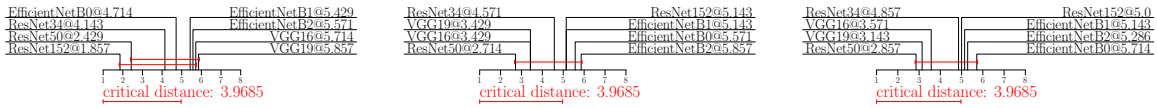
**Figure A3.** Performance of different network architectures in terms of ranking-based evaluation measures for fine-tuned features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.



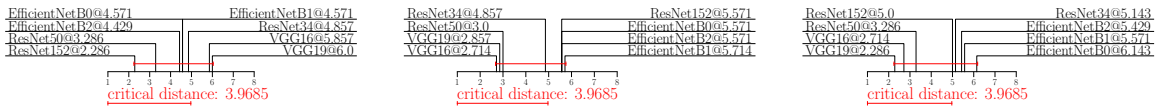

(**a**) End-to-end: Micro-F1 (**b**) Random forest: Micro-F1 (**c**) Extra trees: Micro-F1



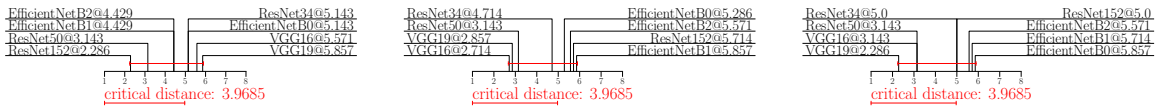

(**d**) End-to-end: Micro-recall (**e**) Random forest: Micro-recall (**f**) Extra trees: Micro-recall



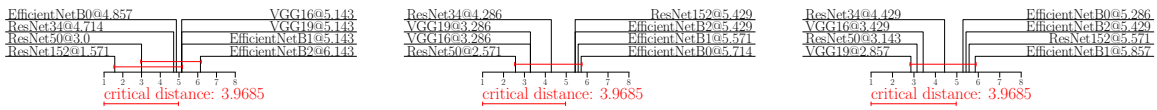

(**g**) End-to-end: Micro-precision (**h**) Random forest: Micro-precision (**i**) Extra trees: Micro-precision




(**j**) End-to-end: Macro-F1 (**k**) Random forest: Macro-F1 (**l**) Extra trees: Macro-F1

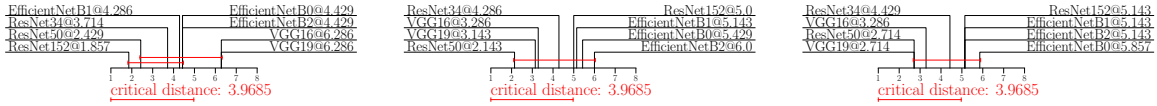



(**m**) End-to-end: Macro-recall (**n**) Random forest: Macro-recall (**o**) Extra trees: Macro-recall

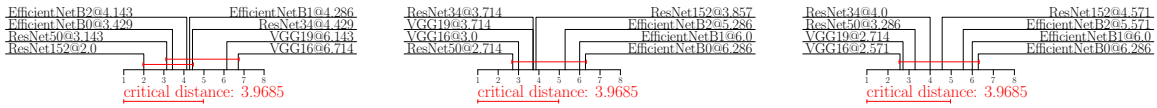



(**p**) End-to-end: Macro-precision (**q**) Random forest: Macro-precision (**r**) Extra trees: Macro-precision

**Figure A4.** Performance of different network architectures in terms of label-based evaluation measures for pre-trained features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

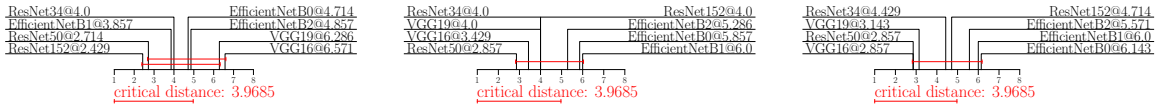



(**a**) End-to-end: Hamming loss (**b**) Random forest: Hamming loss (**c**) Extra trees: Hamming loss

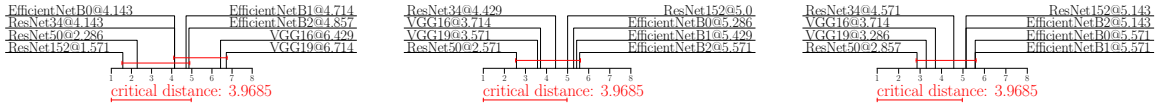



(**d**) End-to-end: Subset Accuracy (**e**) Random forest: Subset Accuracy (**f**) Extra trees: Subset Accuracy

**Figure A5.** Performance of different network architectures in terms of example-based evaluation measures for pre-trained features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

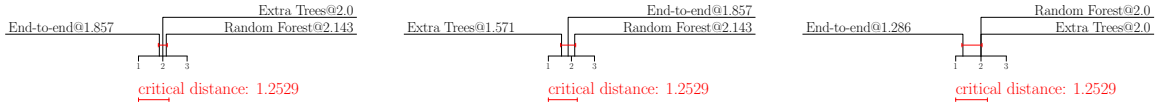



(**a**) End-to-end: Coverage (**b**) Random forest: Coverage (**c**) Extra trees: Coverage

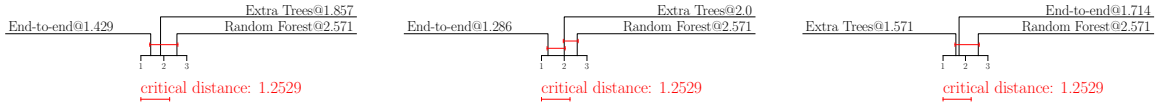



(**d**) End-to-end: Average Precision (**e**) Random forest: Average Precision (**f**) Extra trees: Average Precision

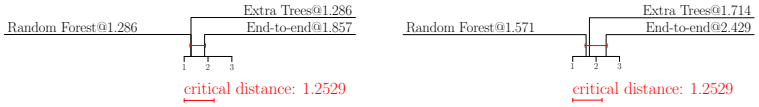
**Figure A6.** Performance of different network architectures in terms of ranking-based evaluation measures for pre-trained features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.



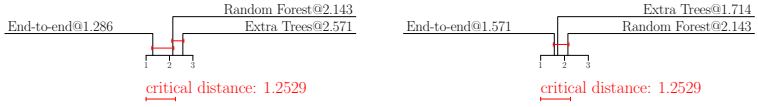

(**a**) Micro $F_1$ (**b**) Micro Precision (**c**) Micro Recall




(**d**) Macro $F_1$ (**e**) Macro Recall (**f**) Macro Precision




(**g**) Hamming loss (**h**) Subset Accuracy




(**i**) Coverage (**j**) Average Precision

**Figure A7.** Performance of different learning methods in terms of all MLC evaluation measures for fine-tuned features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

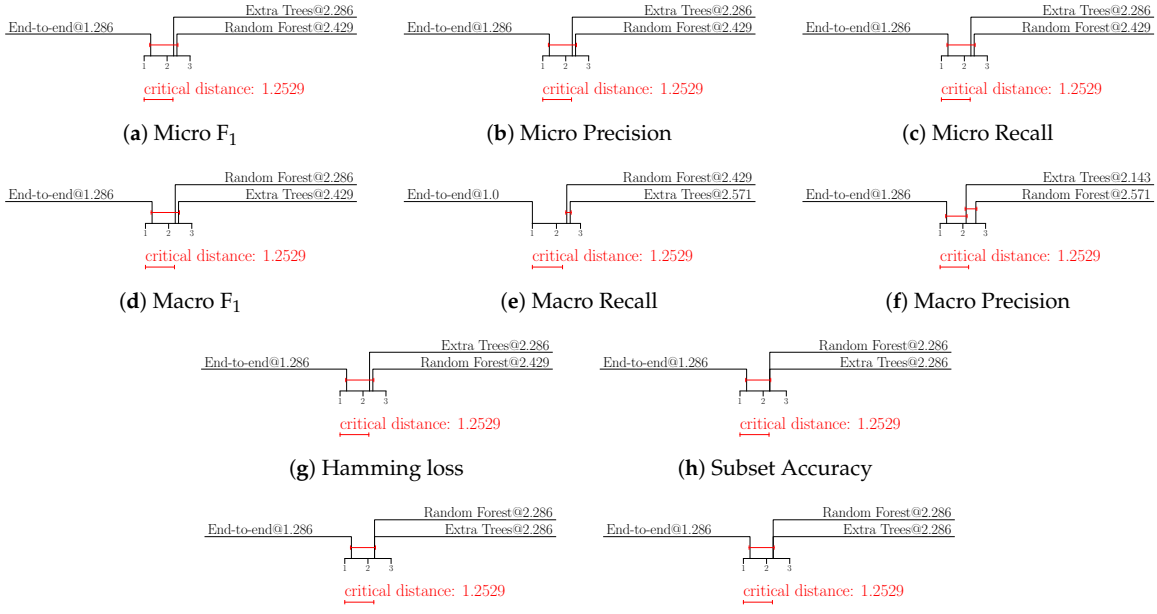



**Figure A8.** Performance of different learning methods in terms of all MLC evaluation measures for pre-trained features. The results are presented in the form of average rank diagrams at a 0.05 significance level. The best ranking methods are at the left-most side of the diagram. The difference among the methods connected with a red line is not statistically significant.

## References

1. Ibrahim, S.K.; Ziedan, I.E.; Ahmed, A. Study of Climate Change Detection in North-East Africa Using Machine Learning and Satellite Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11080–11094. [CrossRef]
2. Chen, H.; Qi, Z.; Shi, Z. Remote Sensing Image Change Detection With Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
3. Ortega Adarme, M.; Queiroz Feitosa, R.; Nigri Happ, P.; Aparecido De Almeida, C.; Rodrigues Gomes, A. Evaluation of Deep Learning Techniques for Deforestation Detection in the Brazilian Amazon and Cerrado Biomes From Remote Sensing Imagery. *Remote Sens.* **2020**, *12*. [CrossRef]
4. Park, M.; Tran, D.Q.; Jung, D.; Park, S. Wildfire-Detection Method Using DenseNet and CycleGAN Data Augmentation-Based Remote Camera Imagery. *Remote Sens.* **2020**, *12*, 910. [CrossRef]
5. Zhang, Q.; Ge, L.; Zhang, R.; Metternicht, G.I.; Liu, C.; Du, Z. Towards a Deep-Learning-Based Framework of Sentinel-2 Imagery for Automated Active Fire Detection. *Remote Sens.* **2021**, *13*, 4790. [CrossRef]
6. Papoutsis, I.; Bountos, N.I.; Zavras, A.; Michail, D.; Tryfonopoulos, C. Benchmarking and scaling of deep learning models for land cover image classification. *ISPRS J. Photogramm. Remote Sens.* **2023**, *195*, 250–268. [CrossRef]
7. Yansheng, L.; Ruixian, C.; Yongjun, Z.; Mi, Z.; Ling, C. Multi-Label Remote Sensing Image Scene Classification by Combining a Convolutional Neural Network and a Graph Neural Network. *Remote Sens.* **2020**, *12*, 4003.
8. Bogatinovski, J.; Todorovski, L.; Džeroski, S.; Kocev, D. Comprehensive comparative study of multi-label classification methods. *Expert Syst. Appl.* **2022**, *203*, 117215. [CrossRef]
9. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), virtual, 3–7 May 2021.
10. Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai, L.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
11. Dimitrovski, I.; Kitanovski, I.; Kocev, D.; Simidjievski, N. Current Trends in Deep Learning for Earth Observation: An Open-source Benchmark Arena for Image Classification. *arXiv* **2022**, arXiv:2207.07189.
12. Pires de Lima, R.; Marfurt, K. Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis. *Remote Sens.* **2020**, *12*, 86. [CrossRef]

13. Khaleghian, S.; Ullah, H.; Kræmer, T.; Hughes, N.; Eltoft, T.; Marinoni, A. Sea Ice Classification of SAR Imagery Based on Convolution Neural Networks. *Remote Sens.* **2021**, *13*, 1734. [CrossRef]
14. Wang, A.X.; Tran, C.; Desai, N.; Lobell, D.; Ermon, S. Deep Transfer Learning for Crop Yield Prediction with Remote Sensing Data. In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies, San Jose, CA, USA, 20–22 June 2018; Association for Computing Machinery: New York, NY, USA, 2018; COMPASS'18.
15. Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; Xu, W. CNN-RNN: A Unified Framework for Multi-label Image Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2285–2294.
16. Chen, Z.; Wei, X.; Wang, P.; Guo, Y. Multi-Label Image Recognition with Graph Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5172–5181.
17. Sumbul, G.; Charfuelan, M.; Demir, B.; Markl, V. BigEarthNet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding. *IEEE Int. Geosci. Remote Sens. Symp.* **2019**, *12*, 5901–5904.
18. Yessou, H.; Sumbul, G.; Demir, B. A Comparative Study of Deep Learning Loss Functions for Multi-Label Remote Sensing Image Classification. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Waikoloa, HI, USA, 26 September–2 October 2020;
19. Sumbul, G.; Kang, J.; Demir, B. Deep Learning for Image Search and Retrieval in Large Remote Sensing Archives. *arXiv* **2020**, arXiv:2004.01613.
20. Hua, Y.; Mou, L.; Zhu, X. Relation Network for Multi-label Aerial Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 4558–4572. [CrossRef]
21. Sumbul, G.; Demİr, B. A Deep Multi-Attention Driven Approach for Multi-Label Remote Sensing Image Classification. *IEEE Access* **2020**, *8*, 95934–95946. [CrossRef]
22. Wang, X.; Duan, L.; Ning, C. Global Context-Based Multilevel Feature Fusion Networks for Multilabel Remote Sensing Image Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11179–11196. [CrossRef]
23. Karen, S.; Andrew, Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
24. Kaiming, H.; Xiangyu, Z.; Shaoqing, R.; Jian, S. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
25. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 6105–6114.
26. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
27. Chaudhuri, B.; Demir, B.; Chaudhuri, S.; Bruzzone, L. "Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1144–1158. [CrossRef]
28. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Lu, X.; Zhang, L. AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
29. Ömrüuzun, F.; Demir, B.; L. Bruzzone, L.; Çetin, Y. Content based hyperspectral image retrieval using bag of endmembers image descriptors. In Proceedings of the 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Los Angeles, CA, USA, 21–24 August 2016; pp. 1–4.
30. Hua, Y.; Mou, L.; Zhu, X. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. *ISPRS J. Photogramm. Remote Sens.* **2019**, *149*, 188–199. [CrossRef]
31. Qi, Q.X.; Panpan, Z.; Yuebin, W.; Liqiang, Z.; Junhuan, P.; Mengfan, W.; Jialong, C.; Xudong, Z.; Ning, Z.; P.M.Takis. MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 337–350. [CrossRef]
32. Sumbul, G.; d. Wall, A.; Kreuziger, T.; Marcelino, F.; Costa, H.; Benevides, P.; Caetano, M.; Demir, B.; Markl, V. BigEarthNet-MM: A Large Scale Multi-Modal Multi-Label Benchmark Archive for Remote Sensing Image Classification and Retrieval. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 174–180. [CrossRef]
33. Kocev, D.; Vens, C.; Struyf, J.; Džeroski, S. Tree ensembles for predicting structured outputs. *Pattern Recognit.* **2013**, *46*, 817–833. [CrossRef]
34. Kocev, D.; Ceci, M.; Stepisnik, T. Ensembles of extremely randomized predictive clustering trees for predicting structured outputs. *Mach. Learn.* **2020**, *109*, 2213–2241. [CrossRef]
35. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]
36. Xiao, Q.; Liu, B.; Li, Z.; Ni, W.; Yang, Z.; Li, L. Progressive Data Augmentation Method for Remote Sensing Ship Image Classification Based on Imaging Simulation System and Neural Style Transfer. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 9176–9186. [CrossRef]

37. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
38. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

# Chapter 4

# Semi-Supervised Multi-Label Classification of Remote Sensing Images With Predictive Clustering Trees and Ensembles

This chapter presents our contributions to semi-supervised learning (SSL) for remote sensing image classification using predictive clustering trees (PCTs) and their ensembles. It contains two closely related studies. The first introduces a semi-supervised framework for multi-class and multi-label classification of remote sensing images using SSL-PCTs and SSL-RForests. The second extends this framework with PCA-based dimensionality reduction to address the computational challenges associated with high-dimensional CNN feature representations. Together, these studies demonstrate how SSL can effectively enhance predictive performance and efficiency for structured output prediction in remote sensing.

From the hypotheses defined in Section 1.3, this chapter addresses the following:

**H2** Extending supervised methods for structured output prediction (SOP) to a semi-supervised setting improves model performance by effectively leveraging unlabeled data while preserving structured label dependencies in image classification.

**H3** Semi-supervised methods that integrate deep feature extraction from images with predictive clustering frameworks outperform their supervised counterparts in multi-label image classification (MLC) tasks.

Our results confirm **H2** and **H3** for both MCC and MLC tasks. SSL-PCTs and SSL-RForests consistently outperform supervised variants, and PCA-based dimensionality reduction further enhances computational efficiency without sacrificing predictive performance.

## 4.1 Semi-Supervised MLC of RSIs With PCTs and Ensembles

We propose an SSL framework for multi-class and multi-label classification of remote sensing images that combines SSL decision trees with discriminative CNN-based feature representations. The framework integrates predictive clustering trees (SSL-PCTs) and random

forests (SSL-RForests) trained on deep CNN embeddings to leverage both labeled and unlabeled data.

A comprehensive experimental study was conducted on multiple RSI datasets, evaluating different CNN backbones, SSL strategies, and statistical comparisons. The results show that SSL methods, particularly SSL-PCTs and SSL-RForests, consistently outperform their supervised counterparts across different labeled data ratios. SSL-RForests achieve the best results when more than 1% of data is labeled, while single SSL-PCTs show the largest relative improvement when labeled data are scarce.

We also analyzed end-to-end SSL methods and found that their effectiveness depends strongly on the underlying network architecture, with EfficientNet performing best across all experiments. These findings confirm that incorporating unlabeled data through SSL enhances performance and robustness in structured image classification.

The paper included in this section is:

- Stoimchev, M., Levatić, J., Kocev, D., and Džeroski, S. (2024), "Semi-Supervised Multi-Label Classification of Land Use/Land Cover in Remote Sensing Images with Predictive Clustering Trees and Ensembles", in *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1–16, 2024.

**The contributions of Marjan Stoimchev to this paper are as follows.** Marjan Stoimchev designed and implemented the semi-supervised learning framework, developed the convolutional autoencoder (CAE) for feature extraction, and integrated it with SSL-PCTs and SSL-RForests. He performed all experiments, analyzed the results, and prepared and revised the manuscript in collaboration with the co-authors.

**Code availability.**

- **Main framework:** Implementation of the semi-supervised framework, including SSL-PCT and SSL-RForest models, training pipelines, evaluation scripts, and the convolutional autoencoder (CAE) used for feature extraction — available at `https://github.com/marjanstoimchev/SSLMLC_RemoteSensing`.
- **Clus framework:** The experiments are based on the *Clus* predictive clustering framework — available at `https://github.com/knowledge-technologies/clus`.

IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 62, 2024 4706416

# Semi-Supervised Multi-Label Classification of Land Use/Land Cover in Remote Sensing Images With Predictive Clustering Trees and Ensembles

Marjan Stoimchev, Jurica Levatić, Dragi Kocev, and Sašo Džeroski

***Abstract*— The task of remote sensing image (RSI) classification has been studied extensively in the geoscience and remote sensing (RS) community. While deep learning methods have shown great success in solving this task, their reliance on large-scale labeled datasets is a serious limitation when dealing with complex labels and multiple semantic categories. The process of annotating such datasets can be time-consuming and tedious, leading to limited availability of labeled data and reduced performance of supervised learning methods. To address this issue, semi-supervised learning (SSL) methods can be applied, as they use both the limited labeled data and the abundant unlabeled data. In this article, we propose an effective SSL framework for RSI classification, which combines two key concepts. First, we employ a deep convolutional feature extractor to learn feature representations that encode the images into a lower dimensional feature space, capturing the rich semantic context present in RSI. Second, we utilize semi-supervised predictive clustering trees (PCTs) and ensembles thereof to learn from both the labeled and unlabeled data. To evaluate the effectiveness of the proposed framework, we compare it against several state-of-the-art self-supervised and semi-supervised methods from the literature. We conduct extensive experiments on ten publicly available land use/land cover RSI classification datasets: five for multi-class classification (MCC) and five for multi-label classification (MLC). The results demonstrate that the proposed framework has superior predictive performance compared to state-of-the-art methods from the literature, highlighting its effectiveness in semi-supervised RSI classification.**

***Index Terms*— Convolutional autoencoders (CAEs), multiclass classification (MCC), multi-label classification (MLC), remote sensing (RS) images, self-supervised learning, semi-supervised learning (SSL), tree ensembles.**

Manuscript received 29 May 2024; accepted 5 July 2024. Date of publication 11 July 2024; date of current version 1 August 2024. This work was supported in part by European Commission through the ASSAS Project under Grant 101059682, the ELIAS Project under Grant 101120237, the INQUIRE Project under Grant 101057499, the PARC Project 101057014, and the TAILOR Project under Grant 952215; and in part by Slovenian Research Agency through the Research Program P2-0103 under Project J1-3033, Project J2-2505, Project J2-4452, Project J2-4660, Project J3-3070, Project J4-3095, Project J5-4575, Project J7-4636, Project J7-4637, and Project N2-0236. *(Corresponding author: Marjan Stoimchev.)*

Marjan Stoimchev is with the Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia, and also with the Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia (e-mail: marjan.stoimchev@ijs.si).

Jurica Levatić, Dragi Kocev, and Sašo Džeroski are with the Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia (e-mail: jurica.levatic@ijs.si; dragi.kocev@ijs.si; saso.dzeroski@ijs.si).

Digital Object Identifier 10.1109/TGRS.2024.3426981

## I. Introduction

REMOTE sensing is a valuable tool for studying the Earth's surface and its phenomena from a distance. It allows us to collect data without the need for physical contact with the objects of interest. In the analysis of remote sensing image (RSI) data, machine learning techniques are often employed to identify patterns and extract meaningful information, particularly when dealing with large-scale image datasets [1].

There are two common machine learning tasks in the analysis of RSIs: multiclass classification (MCC) and multi-label classification (MLC). In the older and more prevalent task of MCC, a single label is associated with an image. An example of MCC would be classifying different types of crops in agriculture. The limitation of MCC is that it assumes that an RSI belongs to a single class. However, in real-world situations, multiple classes can be present in a single image. This limitation can be addressed by using MLC, which allows for a more flexible representation of the inherent complexity present in the real world and avoids the oversimplification of the classification process. By considering multiple labels for a single RSI, MLC can lead to more accurate and meaningful results, especially when dealing with complex and heterogeneous landscapes.

The creation of large-scale labeled datasets has allowed deep learning models to surpass the performance of traditional machine learning methods, especially in the fields of image recognition and classification. As a result, researchers from various disciplines, including geoscience and remote sensing (RS), are exploring the potential of deep learning in addressing complex problems in their respective fields [2], [3], [4], [5], [6], [7], [8], [9]. Such approaches can automatically extract knowledge and learn discriminative representations from data in an end-to-end manner and in less-constrained environments. However, obtaining domain knowledge from field experts, in the form of annotation of high volumes of image data, is a tedious and time-consuming task. It can lead to scarcity of labeled data and consequently limited predictive performance of supervised methods.

This is especially true for the domain of RS because of several reasons. First, the complexity of the visual context present in the images and the correlation among the labels are relatively high (e.g., predicting composite concepts in the images). Second, RS imagery might comprise vast geograph-

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

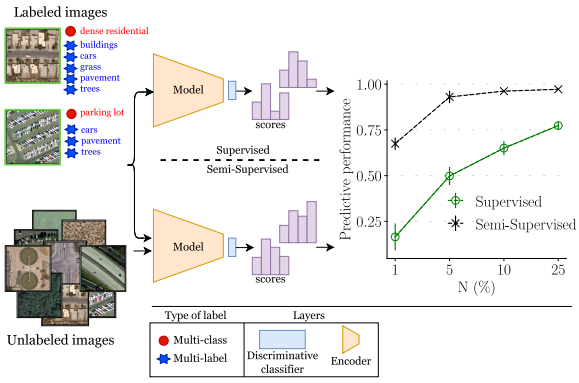



Fig. 1. Schematic illustration of the supervised and SSL settings for multiclass and multi-label RS image classification.

ical areas with some of them remote and difficult to access (e.g., polar and mountainous regions are difficult to access, and conditions can change rapidly; hence, data collection in these environments is both expensive and dangerous). Third, there is a need for specialized knowledge and equipment for accurate labeling (e.g., soil and land degradation assessments require field surveys for soil properties and need specialized equipment and experts). Furthermore, RS imagery is used to observe dynamic phenomena that result in rapid and often unpredictable changes in environmental conditions (e.g., real-time monitoring of natural disasters, such as earthquakes, tsunamis, and floods; monitoring of agricultural landscapes is challenging due to factors such as weather conditions, irrigation and fertilization practices, and pest infestation). Overall, labeled data are indispensable for the success of predictive models using remote sensing imagery, as they enhance the accuracy, reliability, and generalizability of these models, thus improving their application in downstream RS tasks. However, as mentioned earlier, obtaining labeled data is challenging.

There are several approaches for tackling the problem of labeled data availability [10]. The most common one is to exploit proven deep learning models trained on large-scale image datasets, such as ImageNet [11], and to fine-tune them on limited-size datasets for the downstream task. While directly transferring the learned knowledge from ImageNet to RSI can be a successful and easy solution [12], it still requires a significant amount of labeled images of the target domain (typically in the thousands). Another family of methods tries to address the small-sample problem through domain adaptation, where the general idea is to align the distribution of the source and target domains [13]. Significant successes have been achieved with self-supervised learning approaches that focus on various pretext tasks for pretraining, such as image reconstruction by using autoencoders [14], [15], [16] and modeling image similarity or dissimilarity between two or more views through the contrastive learning paradigm [17]. To address the problem of limited labeled data, in this article, we focus on semi-supervised learning (SSL), which leverages both unlabeled and labeled and, hence, can provide greater generalisability [10].

The use of SSL methods reduces the need for tedious manual labeling and can help SSL methods to achieve better performance than fully supervised methods that use only labeled data. The general concept of SSL is illustrated in Fig. 1. Several SSL solutions have been successfully applied in a number of domains, such as image classification [18], object detection [19], and semantic segmentation [20]. However, to the best of our knowledge, the existing SSL approaches for RSI classification primarily focus on the multiclass setting [21], [22], [23], [24], [25], and there is a lack of SSL methods able to tackle both multiclass and multi-label RSI classification tasks. As discussed, RSIs are intrinsically multi-label, and moreover, obtaining multiple labels is even more complex and time-consuming. This exacerbates the problem of labeled data scarcity and emphasizes the need for SSL methods in the MLC of RSIs.

Among the various methods in the realm of SSL, the most dominant ones include consistency regularization-based approaches [18], [26], [27], pseudolabeling [28], [29], self-training [30], and cotraining [31], [32]. Consistency regularization methods involve perturbing input images using random augmentations to establish an invariant output distribution. Pseudolabeling and self-training rely on using a model trained on a small set of labeled examples to generate new predictions based on unlabeled data, creating so-called pseudolabels, typically derived using confidence scores. Similarly, cotraining methods train two or more different models on the same dataset. The models subsequently share their predictions with one another, and only the samples on which they agree are labeled with pseudolabels and utilized for training the next iteration of the models. Another set of SSL methods is based on contrastive learning [33], [34], [35], which involves training a model to determine whether two samples are similar or dissimilar. This is done by maximizing the agreement between two representations of the same sample, subjected to different perturbations.

In addition, there are SSL methods based on generative modeling, where models are trained to generate new data that are similar to a given dataset. Such approaches have a range of potential applications, such as generating new images or text or creating synthetic data to expand a small labeled dataset. One well-known method of generative modeling is the approach of generative adversarial networks (GANs) [36], [37]. In this approach, two neural networks are trained in an adversarial fashion: a generator network tries to deceive the discriminator network that aims to correctly differentiate between real and generated images. Another approach to generative modeling is the use of variational autoencoders (VAEs) [38]. VAEs are a type of neural network that can learn to encode data into a lower dimensional representation and then decode the representation back into the original data. VAEs can be used to generate new data by sampling from the learned representation and then decoding the samples into the original data space. Despite the apparent need for SSL, to the best of our knowledge, there is very limited research on SSL for multi-label RSI classification, with only a few such methods available [39], [40] to solve the task at hand.

To address the lack of semi-supervised methods for multi-label RSI classification, in this article, we propose an effective framework that combines ideas from deep learning on the one hand and SSL based on predictive clustering trees (PCTs) and random forest ensembles thereof [41], [42] on the other hand. This framework integrates the advantages of both approaches: 1) learning feature representations using deep convolutional-based feature extractors that capture the rich semantic context of RSIs and 2) leveraging a small amount of labeled data with abundant unlabeled data in an SSL fashion. The latter (PCTs and their ensembles) can address both MCC and MLC tasks, considering the label dependencies inherent in MLC tasks. Note that the proposed framework is general and can be readily applied to other domains facing a scarcity of labeled images.

The main contributions of our work can be summarized as follows.

1) We propose an SSL framework for multi-label and multiclass RSI classification, which combines decision trees for SSL with the discriminative feature representations learned by a backbone CNN feature extractor.
2) We design and execute a comprehensive experimental analysis of the performance of the proposed framework using multiple RSI datasets for MCC and MLC, various network architectures, and different evaluation measures. We analyze the results quantitatively (through visual inspection of the classification of example images) and statistically (through the Friedman test and Nemenyi posthoc analysis).
3) We show that semi-supervised methods, particularly SSL PCTs and SSL random forests, robustly and consistently outperform their supervised counterparts across the various experimental setups. This indicates the effectiveness of incorporating unlabeled data in improving classification accuracy and its potential for versatile use in RSI classification.
4) We demonstrate that the SSL random forest method is the top performer among the considered state-of-the-art self-supervised and SSL methods, particularly for datasets with more than 1% labeled data. The significant improvement in the predictive performance of SSL over supervised learning was particularly evident in single PCTs. These findings were confirmed through statistical tests and average rank diagrams.
5) We finally show that the semi-supervised convolutional autoencoder (CAE) method shows improvements over the supervised baseline only for certain network architectures, such as EfficientNet. This suggests that it is very important to use the appropriate network architecture for SSL in end-to-end learning scenarios.

## II. Methodology

In this section, we present the proposed approach for semi-supervised MCC and MLC of RSI (see Fig. 2). We describe our two-stage learning method that is composed of a feature extraction and a classification part. The feature extraction uses a standard CNN-based feature extractor to represent images in a lower dimensional space. The classification part employs a learning method for SSL based on PCTs and tree-ensembles for MLC (and MCC).

### A. Feature Extraction

The feature extraction component is a crucial part of our proposed method. Its task is to capture and represent the information present in the images. In this context, a simple CNN backbone feature extractor $E(\mathbf{x}, \theta)$ is used. It takes an RSI $\mathbf{x} \in \mathbb{R}^{w \times h \times 3}$ as input, where $w$ and $h$ represent the image dimensions, 3 represents the three channels of RGB images, and $\theta$ signifies the parameters of the CNN that need to be learned during training. As a result, a $d$-dimensional feature representation $\mathbf{f} \in \mathbb{R}^d$ is generated from the last pooling layer. The CNN backbone is fully convolutional to ensure versatility and applicability to images of varying sizes, as needed in many computer vision tasks, including RSI classification.

### B. Semi-Supervised PCTs and Ensembles

To achieve accurate and efficient classification of RSI with limited labeled data, we utilize PCTs and ensembles of PCTs in an SSL setting for both MCC and MLC tasks [41], [42] (see Fig. 2).

PCTs provide a sophisticated extension of traditional decision trees by allowing predictions of complex, structured output data. PCTs view decision trees as a hierarchical set of clusters, with the initial cluster encompassing all available data. Through recursive division, smaller clusters are formed as one moves from the root to the leaves of the tree. The ensemble version employs PCTs as base models in the ensemble. The ensemble generates predictions for new examples by considering the collective predictions of all PCTs in the ensemble. The SSL PCTs and ensembles differ from these supervised counterparts in that they use both labeled and unlabeled examples in the training process.

The input to the algorithms comprises both labeled and unlabeled examples, in our case, the learned feature representations: $\mathbf{f} = \mathbf{f_l} \cup \mathbf{f_u}$, where $\mathbf{f_l}$ and $\mathbf{f_u}$ are the feature representations generated from the labeled and unlabeled examples, respectively. The tree construction algorithm uses the variance function to evaluate the candidate splits in the tree and choose the best one. The SSL version of the variance function considers both the descriptive and target attributes and can, thus, exploit both labeled and unlabeled examples [41], [42]

$$\mathrm{Var}_f = w \cdot \mathrm{Var}_f(Y) + (1 - w) \cdot \mathrm{Var}_f(X) \tag{1}$$

where $w \in [0, 1]$ balances the contributions of the target space ($Y$) and the descriptive space ($X$) to the variance function $\mathrm{Var}_f$ (note that for unlabeled examples, only the descriptive space is available). The $w$ parameter is key to the flexibility of the learned models, allowing learning to span the entire spectrum from fully supervised ($w = 1$) to fully unsupervised ($w = 0$) models. By controlling the weight given to unlabeled examples using the $w$ parameter, the level of supervision for different datasets can be appropriately adjusted. This acts as a safety mechanism to guard against the performance degeneration

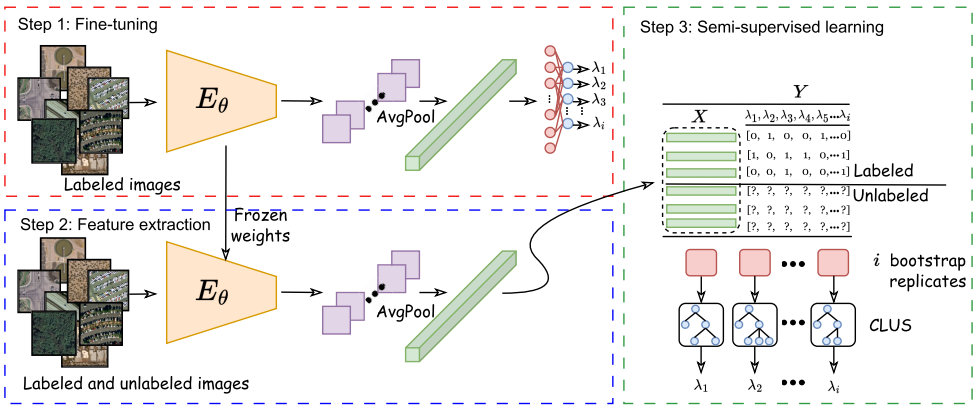



Fig. 2. Overview of the proposed framework for semi-supervised MLC of RSIs. In the first step, a backbone model is fine-tuned on the labeled training set, and in the second step, PCTs are utilized for SSL, where the inputs to the algorithm are labeled and unlabeled learned feature representations. Note that while tree ensembles are depicted, we also consider single PCTs.

caused by the use of unlabeled data [43]. The value of $w$ is selected by inner cross-validation on the labeled set of data. In theory, if an optimal value of $w$ is chosen, semi-supervised PCTs would never perform worse than supervised PCTs since supervised PCTs are a special case of SSL-PCTs, obtained for $w = 1$. However, since $w$ is chosen by internal cross-validation on the training set, the chosen value can be suboptimal for the test set, and hence, the semi-supervised method can perform worse than the supervised one. This, however, rarely happens in practice [41], [42].

The variance of a set of examples $E$ on the target space $Y$ (i.e., a set of $T$ binary labels $Y_1, \ldots, Y_T$) is calculated as follows:

$$\mathrm{Var}_f(E, Y) = \frac{1}{T} \cdot \sum_{i=1}^{T} \mathrm{Gini}(E, Y_i). \tag{2}$$

The variance on the descriptive space $X$ consisting of $D$ attributes of a set of examples $E$ is computed as follows:

$$\mathrm{Var}_f(E, X) = \frac{1}{D} \cdot \sum_{i=1}^{D} \mathrm{Var}(E, X_i) \tag{3}$$

where the variances and the Gini scores are calculated as follows, respectively:

$$\mathrm{Var}(E, X_i) = \frac{\frac{N-1}{K_i-1} \cdot \sum_{j=1}^{K_i} (x_{i,j})^2 - N \cdot \left(\frac{1}{K_i} \cdot \sum_{j=1}^{K_i} x_{i,j}\right)^2}{N}$$

$$\mathrm{Gini}(E, Y_i) = 1 - \left(\sum_{j=1}^{K_i} \frac{|\{e : e \in E \wedge y_{ij} = 1\}|}{K_i}\right)^2 = \left(\sum_{j=1}^{K_i} \frac{|\{e : e \in E \wedge y_{ij} = 0\}|}{K_i}\right)^2 = 2\hat{p}_i(1 - \hat{p}_i)$$

where $N$ is the number of examples (both labeled and unlabeled), $K_i$ is the number of examples with nonmissing values of the $i$th label $Y_i$, and $\hat{p}_i$ is the probability of label $Y_i$, estimated by using only examples for which the value for variable $Y_i$ is known.

## III. Experimental Design

In this section, we provide a comprehensive overview of the experimental design employed in the comparative analysis of the performance of our methods for SSL on RSI data. First, we describe the multiclass and multi-label RSI datasets used to learn the predictive models. Next, we thoroughly explain the evaluation strategy, specifically focusing on the SSL evaluation protocol. We next provide a concise explanation of the metrics used to assess the predictive performance of the trained models for MCC and MLC of RSI. Furthermore, we briefly outline several state-of-the-art semi-supervised and self-supervised learning methods from the literature, which are used in our comparison to emphasize the significance of the methods that we propose. Finally, we specify in detail the parameters of our methods used in the training procedure.

### A. Datasets

We assess the performance of the proposed methods on ten publicly available land use and land cover RSI datasets: five datasets for MCC and five datasets for MLC. We selected these datasets based on their diversity along several dimensions, including image resolution, number of labeled images available, and geographical location. The datasets are available at http://eodata.bvlabs.ai. A short description of the datasets is given in the following, while their quantitative details are given in Table I. Moreover, some typical images with their corresponding class labels for both the multiclass and multi-label datasets are given in Fig. 3.

*1) MCC Datasets:* For the MCC task, we used the datasets OPTIMAL-31 [44], RESISC45 [45], RSSCN7 [46], AID [47], and UCM [48]. OPTIMAL-31 is a dataset consisting of 31 distinct categories for scene classification, with images collected from Google Earth and sized at 256 × 256 pixels. The RESISC45 dataset, created by Northwestern Polytechnical University (NWPU), is a publicly available benchmark for RSI scene classification, comprising 31 500 images at 256 × 256 resolution and covering 45 scene classes. The RSSCN7 dataset contains 2800 scene classification images obtained from Google Earth, with a resolution of 400 × 400.

TABLE I

DESCRIPTION OF THE USED MCC AND MLC RSI DATASETS. $|\mathcal{L}|$ DENOTES THE NUMBER OF POSSIBLE LABELS; CARD DENOTES LABEL CARDINALITY (I.E., AVERAGE NUMBER OF LABELS PER IMAGE); DENS DENOTES LABEL DENSITY (AVERAGE PROPORTION OF IMAGES LABELED WITH A GIVEN LABEL); $N$ IS THE NUMBER OF IMAGES IN THE DATASET, OF WHICH $N_{\text{TRAIN}}$ ARE IN THE TRAIN AND $N_{\text{TEST}}$ IN THE TEST DATASETS; AND $w \times h$ IS THE DIMENSION OF THE IMAGES (IN PIXELS)

| Task | Dataset | Image Type | $\lvert\mathcal{L}\rvert$ | $Card$ | $Dens$ | $N$ | $N_{train}$ | $N_{test}$ | $w \times h$ |
|---|---|---|---|---|---|---|---|---|---|
| MCC | OPTIMAL-31 | Aerial RGB | 31 | 1 | 0.032 | 1,860 | 1,488 | 372 | $256 \times 256$ |
| | UCM | Aerial RGB | 21 | 1 | 0.047 | 2,100 | 1,680 | 420 | $256 \times 256$ |
| | RSSCN7 | Aerial RGB | 7 | 1 | 0.142 | 2,800 | 2,240 | 560 | $400 \times 400$ |
| | AID | Aerial RGB | 30 | 1 | 0.033 | 3,000 | 2,400 | 600 | $600 \times 600$ |
| | RESISC45 | Aerial RGB | 45 | 1 | 0.022 | 31,500 | 25,200 | 6300 | $256 \times 256$ |
| MLC | Ankara | Hyperspectral/Aerial RGB | 29 | 9.120 | 0.536 | 216 | 171 | 45 | $64 \times 64$ |
| | UC Merced Land Use | Aerial RGB | 17 | 3.334 | 0.476 | 2,100 | 1,667 | 433 | $256 \times 256$ |
| | AID Multilabel | Aerial RGB | 17 | 5.152 | 0.468 | 3,000 | 2,400 | 600 | $600 \times 600$ |
| | DFC-15 Multilabel | Aerial RGB | 8 | 2.795 | 0.465 | 3,341 | 2,672 | 669 | $600 \times 600$ |
| | MLRSNet | Aerial RGB | 60 | 5.770 | 0.144 | 109,151 | 87,325 | 21,826 | $256 \times 256$ |

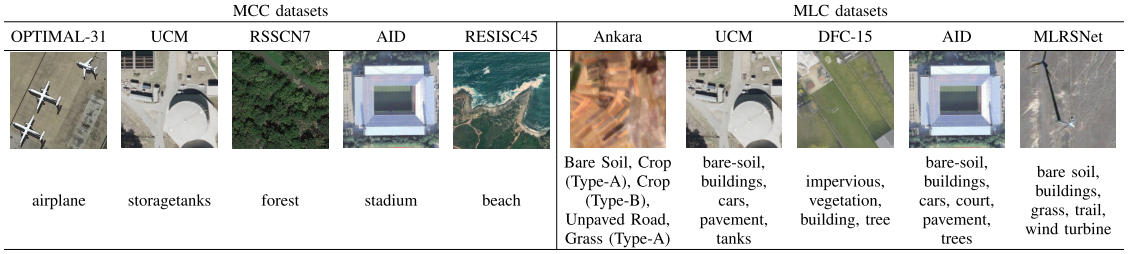



Fig. 3. Illustrative RSIs and their class labels from different multiclass and MLC datasets.

For AID and UCM, we used the multiclass versions of the MLC datasets (as described in the following) with 21 and 30 scene classes, respectively.

*2) MLC Datasets:* For the MLC task, we used the datasets UC-Merced (UCM) Land Use [39], AID [49], Ankara HIS archive [50], DFC-15 [51], and MLRSNet [52]. UCM Land Use contains 2100 images grouped into 21 broad categories at a scene level, with a total of 17 object-level labels. The resolution of the images is $256 \times 256$. AID is a more challenging dataset than UCM, containing 3000 aerial images grouped into 30 object categories, with 17 object-level labels. The image resolution of this dataset is $600 \times 600$. Ankara HIS is a small hyperspectral image archive consisting of only 216 low-resolution images at $63 \times 63$ pixels, acquired by the NASA EO-1 satellite's Hyperion sensor from the area surrounding the city of Ankara in Turkey. DFC-15 is built from a semantic segmentation dataset from the 2015 IEEE GRSS data fusion contest, containing a total of 3342 image patches with a resolution of $600 \times 600$ pixels. The image patches are grouped into eight different object categories. Finally, the MLRSNet dataset is the largest among the MLC datasets, containing 109 161 images annotated into 46 categories, with an image resolution of $256 \times 256$ pixels. Moreover, each image in the dataset is labeled with several of the 60 predefined class labels. In order to ensure that the dataset is comparable in size to the other datasets used in our study, we sampled (using stratification) approximately 5% of the dataset.

### B. Evaluation Strategy

To evaluate the performance of the SSL methods, we use the inductive learning setting of SSL, where our goal is to learn models that can predict labels for new examples that are unseen during training. This means that a separate fixed-sized test set is utilized to assess the final model's performance (see Fig. 10). We use different percentages of labeled data from the training sets to investigate the dependence of model performance on the amount of labeled data. Labeled samples were randomly subsampled from the available training set in the percentages $\{1, 5, 10, 25\}$, while the remaining training samples were used as unlabeled samples (with their labels removed). We repeated the experiments five times using different random initializations for each percentage of unlabeled data. The final predictive performance is averaged across the five runs.

### C. Evaluation Measures

For MCC, we use six performance measures: microaveraged and macroaveraged $F1$, precision, and recall; accuracy; and micro-AUPRC. More details regarding the used measures are given by Wu and Zhou [53]. In the MLC scenario, we employed a range of 13 evaluation measures commonly used in this context. Specifically, we focused on two types of measures for the MLC task: bipartition-based measures (including example- and label-based measures) and ranking-based measures. Example-based measures involve computing the average difference between the true and predicted labels for each data point and then averaging these differences across all the examples in the dataset. We used Hamming loss

and subset accuracy from this group. In contrast, label-based measures evaluate each label individually and then average the performance across all labels. For these measures, we used microaveraged and macroaveraged $F1$, precision, and recall, the average area under the precision–recall curve (AUPRC), and weighted AUPRC. Finally, ranking-based measures compare the predicted ranking of the labels to the ground-truth ranking, where the present labels are ranked before the absent labels. From this group, we used ranking-loss, coverage, and one error. To calculate the measures mentioned above, we use `sckit-learn` [54].

### D. Methods for Comparison

To investigate whether semi-supervised methods benefit from the addition of unlabeled data, we compare the semi-supervised versions of PCTs (SSL-PCT) and random forests (SSL-RForest) methods with their supervised counterparts, i.e., supervised PCTs (SL-PCT) and random forest (SL-RForest). In our experiments, we also consider a CAE approach for end-to-end SSL. The supervised version of this approach is only the encoder part of this structure (denoted as baseline), which is trained on the labeled data only. The idea behind the CAE approach is relatively simple. It operates in two general steps to address the task of SSL: 1) self-supervised pretraining on the unlabeled data by using image reconstruction as a pretext task and 2) fine-tuning the encoder part of the CAE on a small portion of labeled examples for the downstream MCC and MLC tasks by using task dependent discriminative classifiers. This approach has been widely exploited in various computer vision tasks [14], [15], [16], where the main idea is also applicable to the RSI domain [55], [56].

In addition, to demonstrate the relevance of our SSL methods, we compare their predictive performance to that of several state-of-the-art self-supervised methods from the literature: BYOL [57], SimSiam [58], SimCLR [59], BarlowTwins [60], MoCo [61], DINO [62], and SwAV [63]. We adapt these methods for the SSL task by using only the pretrained encoder structure of the networks trained with linear evaluation, which is a common practice for utilizing such methods in an SSL setting [7], [64].

We also consider two state-of-the-art approaches specially designed for the SSL task. The first state-of-the-art method, referred to as HR-S$^2$DML [35], is a deep metric learning architecture originally developed for RSI-MCC. The second method is called MSMatch [23]. MSMatch is an extension of the well-established FixMatch methodology [18], incorporating elements such as consistency regularization and pseudolabeling to effectively address the challenges of SSL. We used MSMatch only for MCC datasets. On the other hand, we modified the HR-S$^2$DML method to operate in the MLC setting. First, instead of the categorical cross-entropy (CE) loss for MCC, we use the binary CE (BCE) loss, which is suitable for the MLC task, and second, for classification, we use the MLC version of the kNN classifier. Note that for the two previously published multi-label SSL methods for RSI [39], [40], the implementations are not publicly available; therefore, we did not include them in our experimental comparison.

### E. Parameter Settings

For the feature extraction, we utilized different backbone CNNs implemented in the `PyTorch` framework [65], namely, VGG [66], ResNet [67], and EfficientNet [68], with weights pretrained on ImageNet. We used a total of eight backbone CNNs for comparison. The networks are modified to accept input images of arbitrary size by replacing the last max-pooling layer with an average-pooling operation using kernel size 1. The resulting $d$-dimensional feature representations are given as follows: 4096 features for VGG-16 and VGG-19, 512 features for ResNet-34, 2048 features for ResNet-50 and ResNet-152, 1280 features for EfficientNet-B0 and EfficientNet-B1, and 1408 features for EfficientNet-B2. The networks are fine-tuned for 25 epochs of training on an NVIDIA A100 (40-GB) GPU, by using the Adam optimizer with a fixed learning rate of $1 \times 10^{-4}$, a minibatch of 128, and a mixed precision of training to speed up the training process and reduce memory consumption. We also applied the same training procedure on the CAEs for end-to-end learning, which follows a symmetrical encoder–decoder structure.

The learned representations by the feature extractors are used as inputs to learn supervised and semi-supervised PCTs and random forests. Single trees are pruned using M5 pruning. The PCTs and the ensembles of PCTs are implemented in the `CLUS+` framework [69] (available at https://github.com/knowledge-technologies/clus). We use 100 unpruned trees to construct random forests, with the feature subset size set to the `sqrt` of the total number of features. For the SSL version, we optimize the parameter $w$ using an internal threefold cross-validation procedure, applied to the labeled portion of the training set. We consider values of the parameter $w$ in the range of 0–1, with 0.1 increments. For the state-of-the-art semi-supervised and self-supervised methods, we use the parameter values recommended by their respective authors.

The experiments were conducted on a computer with an `AMD EPYC 7702 2 GHz` 64-core processor and 1 TB of RAM.

### F. Computational Complexity

The computational complexity of the proposed framework depends on the feature extractor part and the SSL part. The complexity introduced within the feature extractor part varies depending on the chosen architecture. For instance, VGG-16 and VGG-19 require approximately 15.3 and 20 GFLOPS. On the other hand, in the-ResNet based architectures, the complexities range from 3.9 GFLOPS for ResNet-34 to 11 GFLOPS for ResNet-152 [67]. Finally, in EfficientNets, renowned for their efficiency, the requirements are notably lower, with EfficientNet-B0 at 0.39 GFLOPS, EfficientNet-B1 at 0.68 GFLOPS, and EfficientNet-B2 at 1 GFLOPS [68].

The SSL part consists of semi-supervised PCTs and random forests. The complexity of building a single SSL-PCT tree is $\mathcal{O}(DN \log^2 N)) + \mathcal{O}((D+T)ND \log N) + \mathcal{O}(N \log N)$, while the complexity of random forests of SSL-PCTs is bounded by $k(\mathcal{O}(D'N' \log^2 N') + \mathcal{O}((T+D)N'D' \log N'))$, where $D$ is the number of descriptive variables, $T$ is the number of labels,

$N$ is the combined number of unlabeled and labeled training examples, $N'$ is the number of bootstrap samples, $D'$ is the number of features considered at each tree node in random forest, and $k$ is the number of trees [42].

## IV. Results and Discussion

This section presents a comprehensive discussion of the experimental evaluation of the proposed methods for SSL on RSI. The evaluation aims to address several research questions that shed light on the advantages and limitations of the proposed methods. The outcomes of the evaluation are analyzed in relation to the following research questions.

1) Can the proposed SSL methods for RSI classification effectively leverage information from unlabeled data to improve performance?
2) Does the ranking of the considered methods remain consistent across different evaluation measures for MLC and MCC?
3) Is the performance improvement of SSL methods consistent across different datasets, and to what extent does SSL outperform SL?
4) How robust is the performance of the proposed SSL methods to changes in the architecture of the underlying CNN feature extractors?
5) How do the results obtained with the proposed methods compare to those state-of-the-art semi-supervised and self-supervised methods for RSI classification from the literature?

### A. Predictive Performance Using Increasing Amounts of Labeled Data

Here, we present the results in the form of learning curves showing the performance of the models learned with varying percentages of labeled data in order to investigate whether SSL methods can benefit from unlabeled data. The results are presented in terms of AUPRC (for MCC tasks) and AU$\overline{\text{PRC}}$ (for MLC tasks) evaluation measures. Note that although we considered multiple evaluation measures in our experiments, for simplicity and easier interpretation, we report the performance using a single evaluation measure for each task. As analyzed and discussed in Section IV-B, the results remain consistent across the different evaluation measures.

The results for the MCC and MLC datasets are presented in Figs. 4 and 5, respectively, where each row represents a specific backbone architecture, and the columns correspond to the different datasets used in the experiments. For each pair consisting of a dataset and a backbone architecture, the SSL methods are compared with their supervised learning (SL) counterparts (resulting in a single graph in the figure). Overall, the results clearly demonstrate that SSL-PCTs and SSL-RForests consistently outperform their supervised counterparts, SL-PCTs and SL-RForests, highlighting the importance of leveraging unlabeled data. Semi-supervised random forests yield overall the best performance, particularly when dealing with small percentages of labeled data.

Concerning the end-to-end CAEs, improvements in the performance of SSL over SL were observed only for specific architecture types, with the EfficientNet base architectures consistently displaying the largest performance gains. A more detailed analysis of these effects can be found in Section IV-D, where we present a more in-depth explanation of these observations.

### B. Performance of Methods Across Different Evaluation Measures

We next investigate the methods' rankings in terms of the observed predictive performance across the six MCC evaluation measures and 13 MLC evaluation measures used throughout the experiments (as described in Section III-C). To assess the stability and consistency of the rankings, in Fig. 6, we present the distribution of ranks for both SSL and SL methods across the different evaluation measures, separately for MCC and MLC tasks.

An inspection of the results reveals that the ranks are fairly stable across the different percentages of labeled examples, particularly for the MCC task. Therefore, we chose to present the results of the remaining analyses using the AUPRC evaluation measure for MCC and the AU$\overline{\text{PRC}}$ evaluation measure for MLC, as these evaluation measures are independent of classification thresholds and provide a reliable evaluation of performance. We note that the Ankara dataset is an exception, where the results across different evaluation measures are not stable for small amounts of labeled data. This might be due to the fact that the Ankara dataset is the smallest among the datasets used, and experiments conducted with 1% and 5% labeled data use very few labeled images.

The SSL methods based on decision trees (SSL-PCTs and SSL-RFs) consistently outperform their SL counterparts. Among the SSL methods, the RForest approach exhibits the best predictive performance across all datasets and displays a more consistent trend when learning from different percentages of labeled examples. On the other hand, the ranks of the SL and SSL end-to-end methods are more stable at higher percentages of labeled examples, while they exhibit greater variability in ranks at lower percentages.

### C. Relative Performance of SL and SSL Methods Across Different Datasets

In this section, our goal is to explore the extent of improvement achieved by SSL compared to SL methods, as well as to determine if the improvement of SSL over SL depends on the dataset at hand. To accomplish this, we calculate the distribution of the relative differences between the performance of SSL and SL methods across various network architectures. This analysis is performed separately for each dataset. The relative difference is defined as follows:

$$r\Delta_M = \frac{M_{\text{SSL}} - M_{\text{SL}}}{M_{\text{SSL}}} \tag{4}$$

where $r\Delta_M$ is positive if SSL outperforms SL and negative otherwise. In the case of MCC, $M$ is AUPRC, while for MLC, $M$ is AU$\overline{\text{PRC}}$.

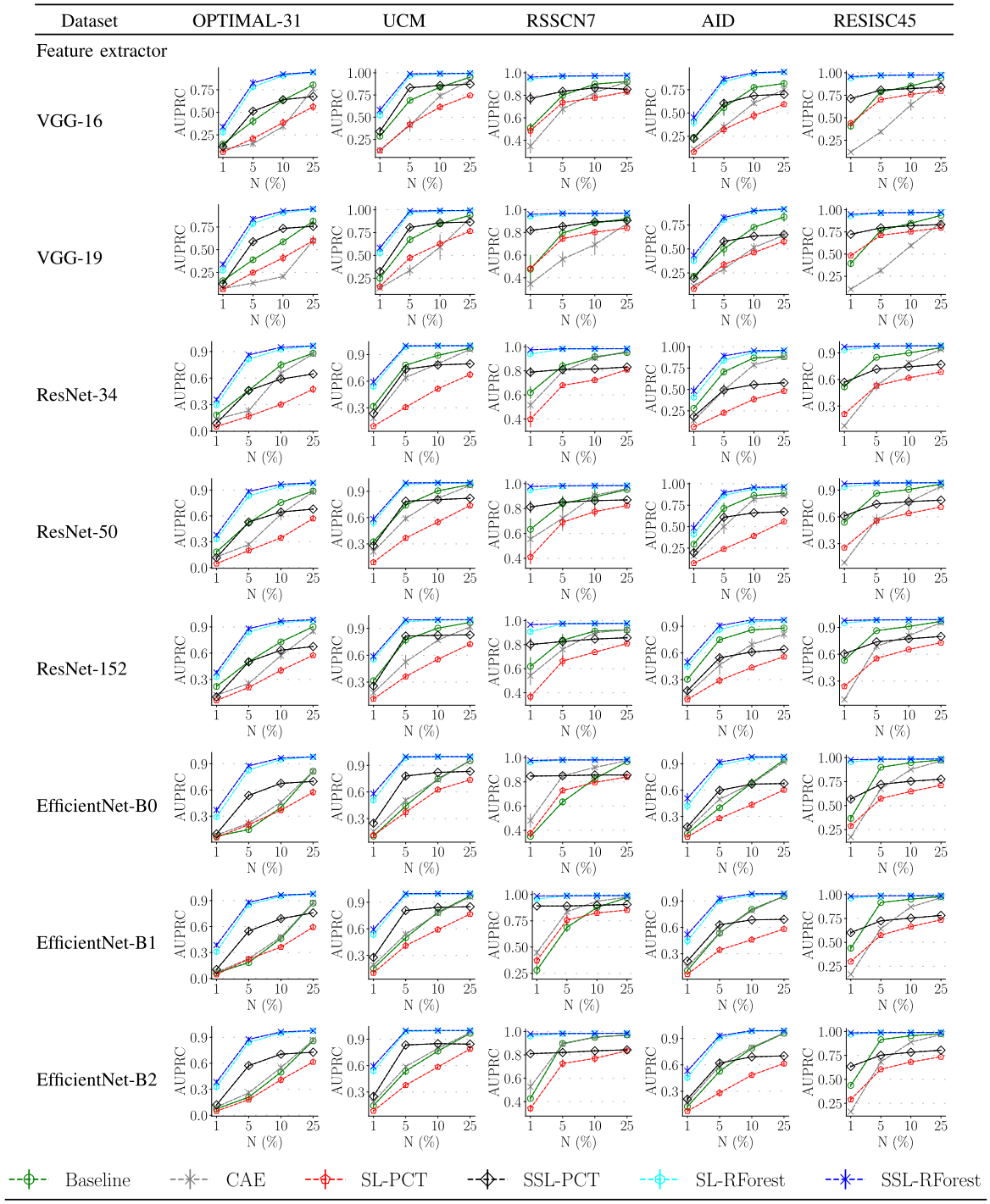



Fig. 4. Predictive performance of SL and SSL on MCC datasets of RSI. PCTs and RForests are applied to features obtained by backbone feature extractors using different CNN architectures. The same CNN architectures are used in end-to-end mode as SL baselines and as parts of the SSL-CAE approach. The learning curves depict $\mu \pm \sigma$ (mean and standard deviation) for AUPRC across five repeats, where higher values indicate better predictive performance. Note that the $\sigma$ values are typically very small and are not visible in the graphs (especially at higher percentages of labeled data).

The distributions of $r\Delta_M$ values are depicted in Fig. 7. Our findings indicate that among the different methods tested, SSL-PCTs and SSL-RForest consistently exhibit improvements over their supervised counterparts across all datasets. Specifically, the most significant improvement in the predictive performance of SSL over SL is observed for the single PCT

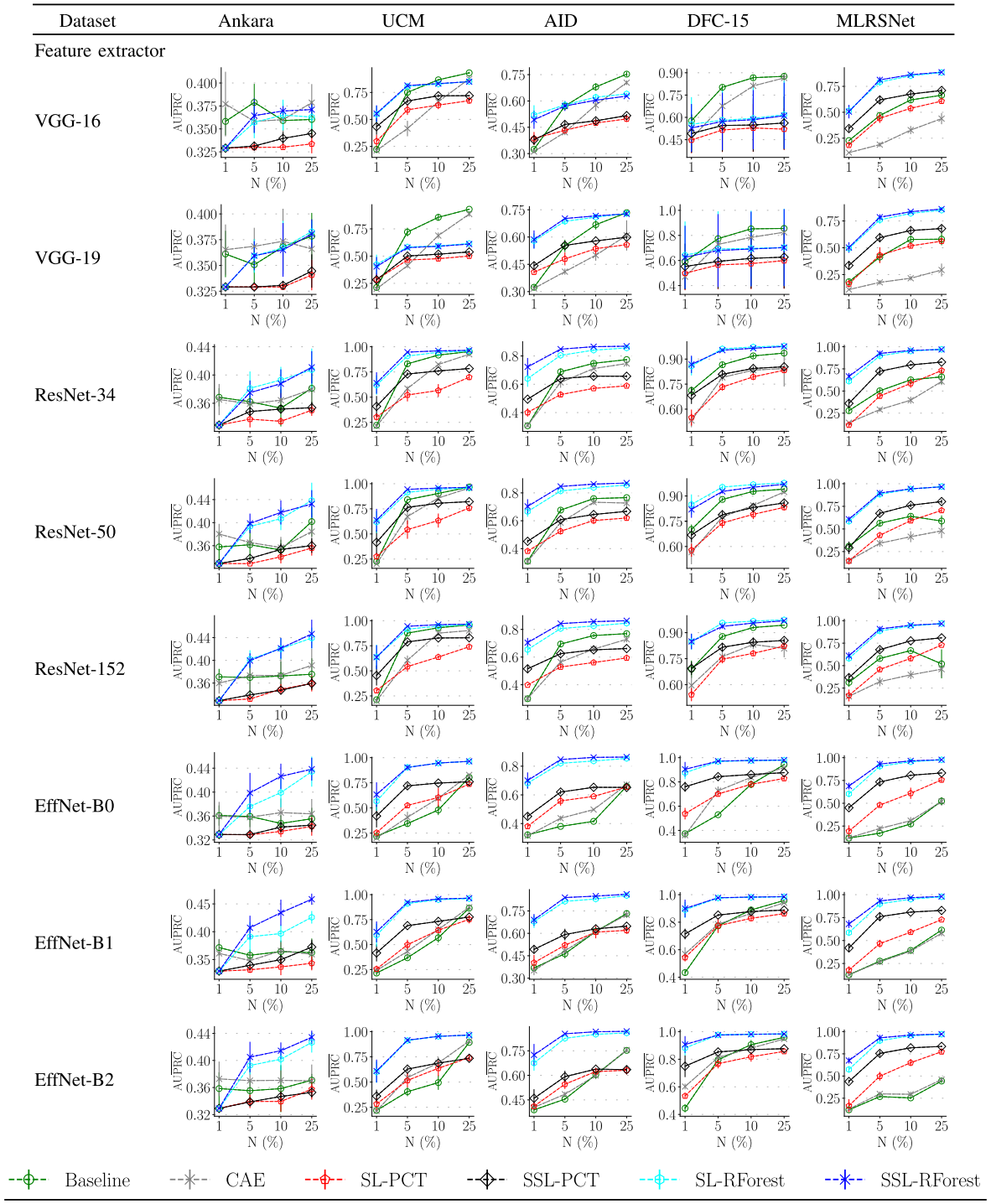



Fig. 5. Predictive performance of SL and SSL on MLC datasets of RSI. PCTs and RForests are applied to features obtained by backbone feature extractors using different CNN architectures. The same CNN architectures are used in end-to-end mode as SL baselines and as parts of the SSL-CAE approach. The learning curves depict $\mu \pm \sigma$ (mean and standard deviation) for $\overline{\text{AUPRC}}$ across five repeats, where higher values indicate better predictive performance. Note that the $\sigma$ values are typically very small and are not visible in the graphs (especially at higher percentages of labeled data).

method (SSL-PCT) compared to its supervised counterpart (SL-PCT), particularly for smaller percentages of labeled data. While SSL-RForest also consistently enhances the predictive performance compared to SL-RForest, the degree of

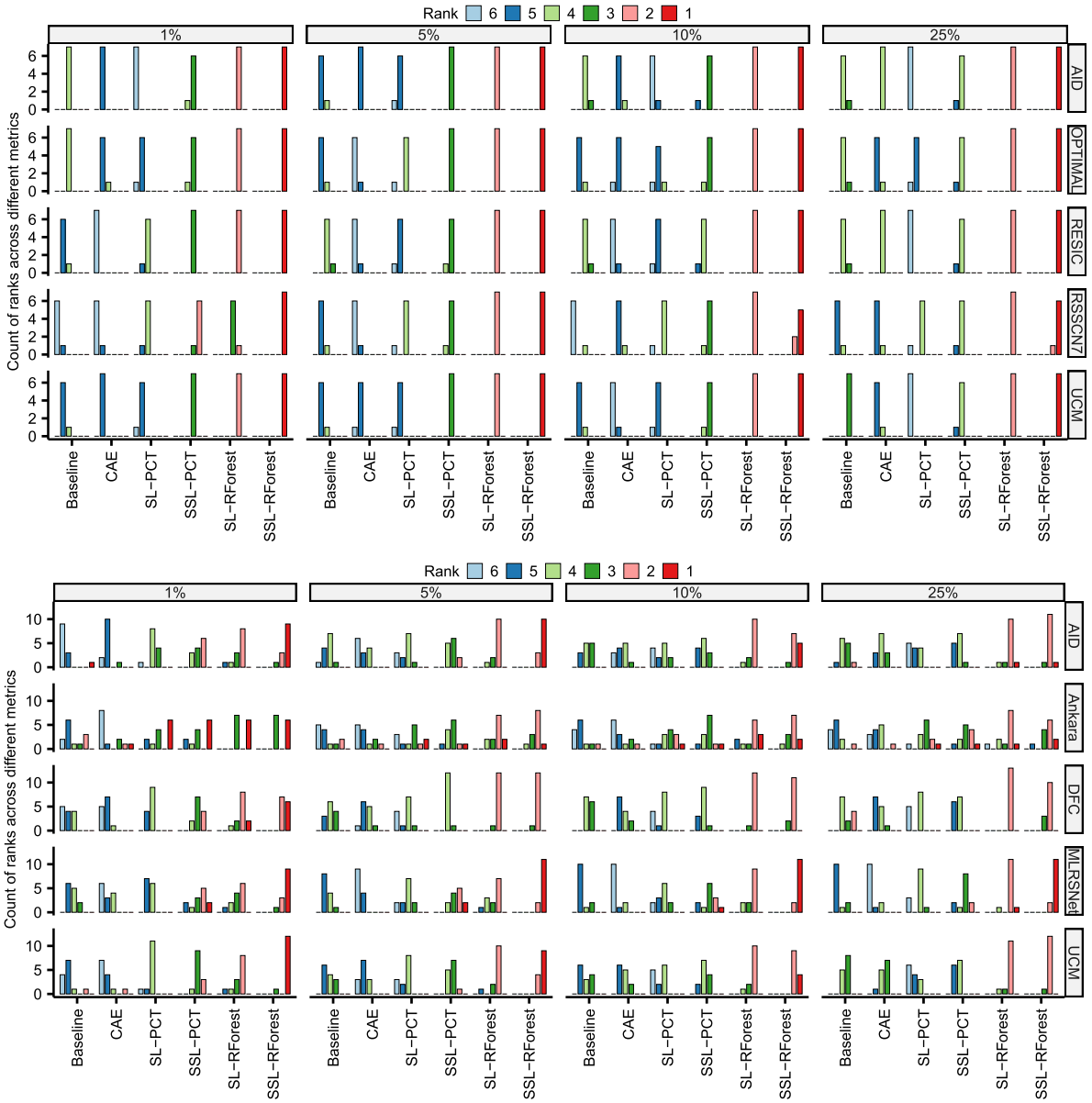



Fig. 6. Distribution of ranks of the newly proposed methods and the competing methods for SL and SSL, across the different MCC and MLC evaluation measures from Section III-C. Each graph represents the counts of ranks for a specific dataset and a given percentage of labeled data taken into consideration while learning the predictive models (ranks were rounded to the nearest integer). Overall, these graphs show the stability and robustness of the proposed methodology under the different evaluation scenarios. The distributions have clear peaks for the MCC datasets while being more spread for the MLC datasets. For the MCC datasets, our methodology is clearly top-ranked across the different datasets and percentages of labeled data. For the MLC datasets, our methodology is most frequently the top-ranked method, albeit not always the first.

improvement is lower than that of SSL-PCTs (although SSL-RForests demonstrate the best overall performance).

On the other hand, CAE does not demonstrate consistent performance improvement over the baseline method—they yield relatively similar results. These results suggest that PCTs and RForest methods are more versatile in using unlabeled data, and their SSL variants yield consistent improvement across different datasets.

### *D. Relative Performance of SL and SSL Methods Across Different Architectures*

In this section, we examine the influence of the network architecture on the (relative) predictive performance of SSL and SL methods. To this end, we calculate the distribution of relative improvement of SSL methods over their supervised counterparts (similarly as in Section IV-C) across different datasets, for each network architecture

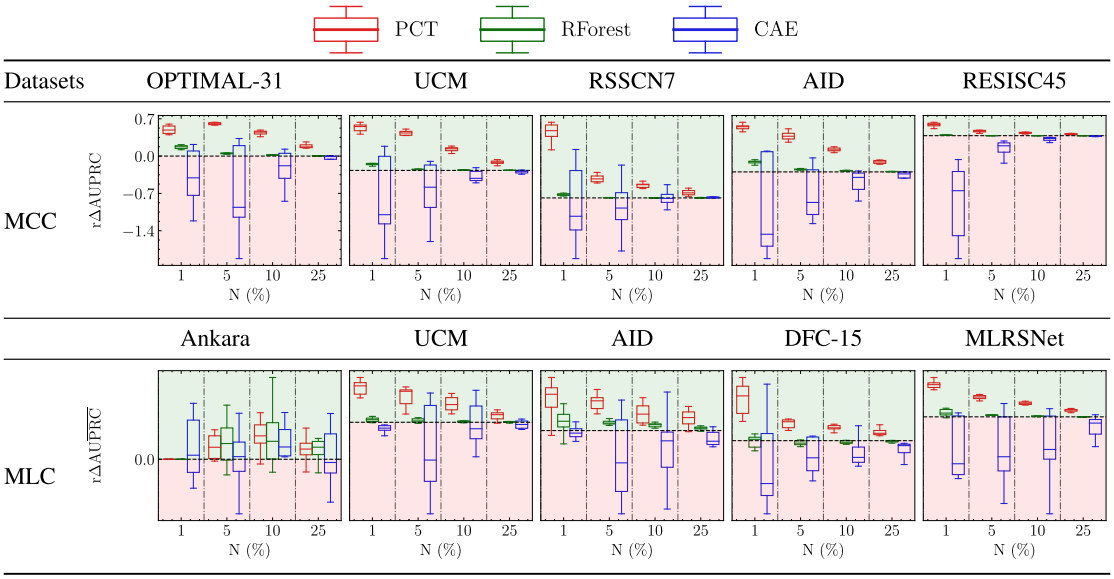



Fig. 7. Improvement of performance of SSL compared to SL in terms of AUPRC/AUPRC for different MCC/MLC datasets. The distribution of the relative difference in performance between the SSL and SL methods across the different network architectures is summarized in the boxplots (one for each method and percentage of labeled data). Regions highlighted in green indicate that SSL outperforms SL, while red indicates the opposite.

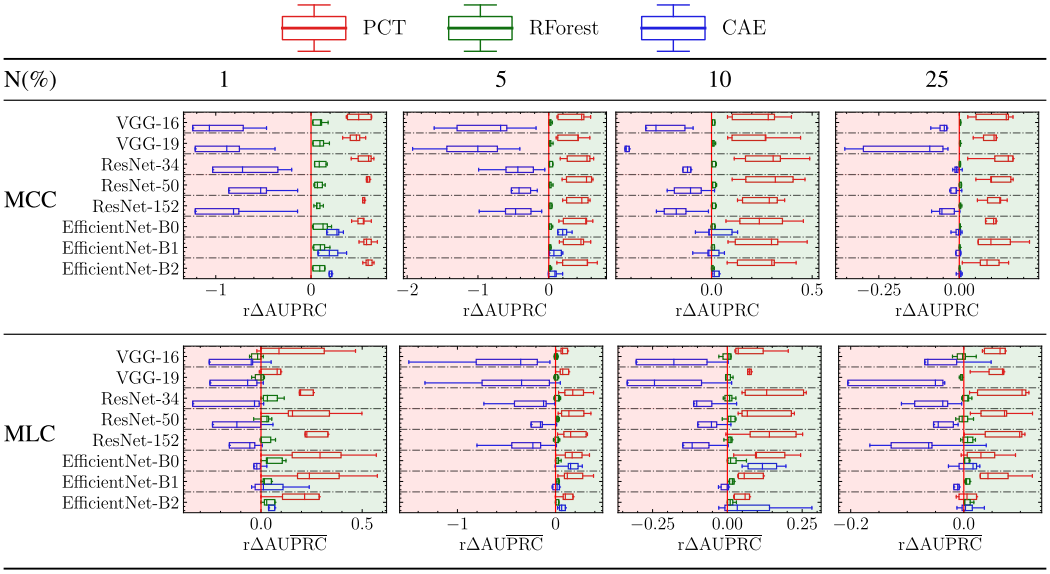



Fig. 8. Improvement of SSL performance compared to SL in terms of AUPRC/AUPRC for various network architectures. The distribution of the relative difference in performance between the SSL and SL methods across the different MCC/MLC datasets is summarized in the boxplots (one for each method and percentage of labeled data). Regions highlighted in green indicate that SSL outperforms SL, while red indicates the opposite.

separately. The results of this analysis are visualized in Fig. 8.

The results demonstrate the consistent superiority of the tree-based semi-supervised methods, namely, SSL-PCTs and SSL-RForests, over their supervised counterparts, regardless of the specific network architecture employed. It is worth noting that SSL-PCTs exhibit a greater relative improvement compared to SSL-RForests.

In contrast, when considering the end-to-end learning method, the semi-supervised CAE method outperforms the supervised baseline only for the EfficientNet-based network architecture. For other network architectures, it actually degrades the performance of the supervised baseline. This suggests that the choice of network architecture plays an important role in achieving better predictive performance with an SSL end-to-end CAE.

TABLE II

PREDICTIVE PERFORMANCE OF THE PROPOSED SEMI-SUPERVISED METHODS (USING THE EFFICIENTNET-B2 NETWORK ARCHITECTURE) COMPARED AGAINST STATE-OF-THE-ART METHODS FROM THE LITERATURE. THE RESULTS ARE PRESENTED USING THE AUPRC MEASURE FOR MCC DATASETS AND THE AUPRC MEASURE FOR MLC DATASETS (HIGHER VALUES INDICATE BETTER PREDICTIVE PERFORMANCE). FOR EACH DATASET AND PERCENTAGE OF LABELED DATA, THE BEST PERFORMANCE IS HIGHLIGHTED IN BOLD

| Dataset | N(%) | Baseline | CAE | MSMatch | HR-$S^2$DML | DINO | SimCLR | MoCo | SimSiam | SwAV | BYOL | BarlowT | SL-PCT | SSL-PCT | SL-RForest | SSL-RForest |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **MCC datasets** | | | | | | | | | | | | | | | | |
| OPT.-31 | 1 | 0.074 | 0.094 | 0.184 | **0.783** | 0.188 | 0.227 | 0.173 | 0.135 | 0.217 | 0.195 | 0.165 | 0.048 | 0.119 | 0.327 | 0.383 |
| | 5 | 0.208 | 0.259 | 0.277 | 0.816 | 0.591 | 0.693 | 0.545 | 0.466 | 0.659 | 0.625 | 0.503 | 0.18 | 0.573 | 0.841 | **0.877** |
| | 10 | 0.497 | 0.55 | 0.676 | 0.859 | 0.685 | 0.784 | 0.639 | 0.452 | 0.785 | 0.713 | 0.597 | 0.409 | 0.705 | 0.948 | **0.962** |
| | 25 | 0.86 | 0.867 | 0.890 | 0.863 | 0.895 | 0.925 | 0.9 | 0.852 | 0.919 | 0.915 | 0.877 | 0.616 | 0.728 | **0.976** | **0.976** |
| UCM | 1 | 0.145 | 0.186 | 0.385 | **0.904** | 0.333 | 0.361 | 0.314 | 0.265 | 0.335 | 0.376 | 0.311 | 0.085 | 0.251 | 0.539 | 0.593 |
| | 5 | 0.537 | 0.588 | 0.645 | 0.926 | 0.814 | 0.85 | 0.74 | 0.595 | 0.833 | 0.801 | 0.733 | 0.377 | 0.834 | 0.986 | **0.997** |
| | 10 | 0.765 | 0.795 | 0.905 | 0.919 | 0.918 | 0.945 | 0.905 | 0.816 | 0.926 | 0.924 | 0.866 | 0.584 | 0.849 | 0.997 | **0.998** |
| | 25 | 0.967 | 0.972 | 0.966 | 0.94 | 0.981 | 0.979 | 0.977 | 0.956 | 0.977 | 0.983 | 0.975 | 0.789 | 0.844 | 0.998 | **0.998** |
| RSSCN7 | 1 | 0.427 | 0.527 | 0.304 | 0.869 | 0.187 | 0.173 | 0.155 | 0.171 | 0.206 | 0.175 | 0.161 | 0.342 | 0.81 | 0.96 | **0.979** |
| | 5 | 0.898 | 0.893 | 0.784 | 0.860 | 0.911 | 0.924 | 0.861 | 0.908 | 0.89 | 0.929 | 0.916 | 0.726 | 0.82 | 0.981 | **0.983** |
| | 10 | 0.951 | 0.953 | 0.813 | 0.897 | 0.953 | 0.946 | 0.944 | 0.942 | 0.933 | 0.957 | 0.951 | 0.77 | 0.835 | **0.984** | **0.984** |
| | 25 | 0.971 | 0.97 | 0.917 | 0.897 | 0.974 | 0.973 | 0.97 | 0.974 | 0.97 | 0.977 | 0.973 | 0.836 | 0.843 | 0.983 | **0.985** |
| AID | 1 | 0.126 | 0.178 | 0.405 | **0.917** | 0.317 | 0.37 | 0.271 | 0.243 | 0.354 | 0.336 | 0.272 | 0.076 | 0.211 | 0.455 | 0.531 |
| | 5 | 0.527 | 0.589 | 0.862 | 0.931 | 0.706 | 0.8 | 0.675 | 0.534 | 0.758 | 0.737 | 0.64 | 0.282 | 0.62 | 0.908 | **0.931** |
| | 10 | 0.788 | 0.799 | 0.963 | 0.934 | 0.871 | 0.914 | 0.856 | 0.711 | 0.878 | 0.898 | 0.832 | 0.484 | 0.691 | 0.979 | **0.988** |
| | 25 | 0.958 | 0.961 | 0.983 | 0.960 | 0.955 | 0.953 | 0.95 | 0.915 | 0.941 | 0.959 | 0.943 | 0.615 | 0.702 | 0.986 | **0.988** |
| RESISC45 | 1 | 0.437 | 0.162 | 0.565 | 0.870 | 0.778 | 0.821 | 0.578 | 0.069 | 0.613 | 0.789 | 0.683 | 0.29 | 0.634 | 0.97 | **0.985** |
| | 5 | 0.915 | 0.683 | 0.891 | 0.885 | 0.928 | 0.938 | 0.912 | 0.401 | 0.885 | 0.944 | 0.918 | 0.605 | 0.752 | 0.987 | **0.989** |
| | 10 | 0.955 | 0.889 | 0.929 | 0.898 | 0.955 | 0.96 | 0.95 | 0.93 | 0.94 | 0.963 | 0.955 | 0.682 | 0.783 | 0.988 | **0.989** |
| | 25 | 0.977 | 0.968 | 0.936 | 0.922 | 0.976 | 0.978 | 0.976 | 0.971 | 0.974 | 0.979 | 0.976 | 0.735 | 0.804 | 0.989 | **0.99** |
| **MLC datasets** | | | | | | | | | | | | | | | | |
| Ankara | 1 | 0.358 | 0.373 | - | **0.394** | 0.358 | 0.370 | 0.386 | 0.37 | 0.365 | 0.378 | 0.348 | 0.329 | 0.329 | 0.329 | 0.329 |
| | 5 | 0.355 | 0.37 | - | 0.347 | 0.378 | 0.359 | 0.387 | 0.385 | 0.371 | 0.364 | 0.374 | 0.339 | 0.339 | 0.393 | **0.405** |
| | 10 | 0.358 | 0.371 | - | 0.350 | 0.378 | 0.369 | 0.376 | 0.383 | 0.363 | 0.38 | 0.363 | 0.339 | 0.347 | 0.402 | **0.414** |
| | 25 | 0.371 | 0.37 | - | 0.388 | 0.365 | 0.376 | 0.368 | 0.37 | 0.37 | 0.375 | 0.386 | 0.358 | 0.353 | 0.427 | **0.434** |
| UCM | 1 | 0.218 | 0.213 | - | **0.785** | 0.477 | 0.483 | 0.418 | 0.407 | 0.467 | 0.48 | 0.453 | 0.282 | 0.36 | 0.599 | 0.609 |
| | 5 | 0.404 | 0.545 | - | 0.801 | 0.808 | 0.844 | 0.759 | 0.695 | 0.836 | 0.802 | 0.741 | 0.516 | 0.628 | 0.912 | **0.915** |
| | 10 | 0.494 | 0.689 | - | 0.853 | 0.881 | 0.91 | 0.86 | 0.806 | 0.891 | 0.893 | 0.851 | 0.637 | 0.688 | **0.953** | **0.953** |
| | 25 | 0.893 | 0.907 | - | 0.893 | 0.949 | 0.953 | 0.948 | 0.931 | 0.95 | 0.952 | 0.94 | 0.742 | 0.736 | **0.964** | **0.966** |
| AID | 1 | 0.389 | 0.404 | - | 0.622 | 0.508 | 0.469 | 0.412 | 0.439 | 0.411 | 0.475 | 0.494 | 0.411 | 0.46 | 0.675 | **0.725** |
| | 5 | 0.454 | 0.485 | - | 0.648 | 0.719 | 0.695 | 0.663 | 0.683 | 0.615 | 0.718 | 0.692 | 0.546 | 0.592 | 0.826 | **0.854** |
| | 10 | 0.604 | 0.599 | - | 0.695 | 0.766 | 0.768 | 0.744 | 0.761 | 0.708 | 0.776 | 0.76 | 0.623 | 0.638 | 0.851 | **0.866** |
| | 25 | 0.752 | 0.755 | - | 0.719 | 0.804 | 0.799 | 0.806 | 0.801 | 0.791 | 0.812 | 0.802 | 0.631 | 0.635 | 0.859 | **0.869** |
| DFC-15 | 1 | 0.445 | 0.602 | - | 0.881 | 0.71 | 0.708 | 0.663 | 0.672 | 0.652 | 0.687 | 0.705 | 0.534 | 0.749 | 0.881 | **0.907** |
| | 5 | 0.796 | 0.813 | - | 0.897 | 0.847 | 0.858 | 0.878 | 0.828 | 0.833 | 0.88 | 0.882 | 0.768 | 0.853 | **0.977** | 0.975 |
| | 10 | 0.906 | 0.88 | - | 0.904 | 0.916 | 0.905 | 0.925 | 0.913 | 0.897 | 0.924 | 0.924 | 0.818 | 0.87 | **0.981** | **0.981** |
| | 25 | 0.959 | 0.947 | - | 0.933 | 0.948 | 0.947 | 0.956 | 0.95 | 0.953 | 0.954 | 0.953 | 0.857 | 0.876 | **0.983** | **0.983** |
| MLRSNet | 1 | 0.119 | 0.128 | - | 0.348 | 0.113 | 0.11 | 0.103 | 0.112 | 0.113 | 0.111 | 0.107 | 0.165 | 0.438 | 0.574 | **0.675** |
| | 5 | 0.268 | 0.298 | - | 0.358 | 0.433 | 0.509 | 0.462 | 0.434 | 0.48 | 0.589 | 0.477 | 0.498 | 0.755 | 0.898 | **0.93** |
| | 10 | 0.253 | 0.295 | - | 0.443 | 0.65 | 0.688 | 0.665 | 0.619 | 0.63 | 0.73 | 0.66 | 0.65 | 0.818 | 0.953 | **0.962** |
| | 25 | 0.443 | 0.46 | - | 0.537 | 0.813 | 0.82 | 0.827 | 0.801 | 0.792 | 0.842 | 0.801 | 0.774 | 0.834 | 0.971 | **0.972** |

### *E. Comparison to State-of-the-Art Methods*

We have conducted a comprehensive performance comparison of the proposed semi-supervised methods against seven state-of-the-art self-supervised methods and two SSL methods for RSI classification (HR-$S^2$DML and MSMatch).

The results of the performance comparison, based on the AUPRC measure for MCC datasets and AUPRC measure for MLC datasets, are presented in Table II. It is evident that the SSL-RForest consistently outperforms all the other state-of-the-art methods, with only a few exceptions. Namely, on a few of the datasets, HR-$S^2$DML is the best performer for 1% of labeled data. The results also demonstrate consistent improvements of SSL-RForest over SL-RForest and SSL-PCT over SL-PCT across all percentages of labeled data and all datasets.

In Fig. 10, we present example predictions for several randomly sampled images from the test sets of the UCM and DFC-15 MLC datasets, with their true labels and labels predicted with different methods. These results indicate that the SSL-RForest method is the most precise among the other methods; it clearly has the fewest false positives and false negatives.

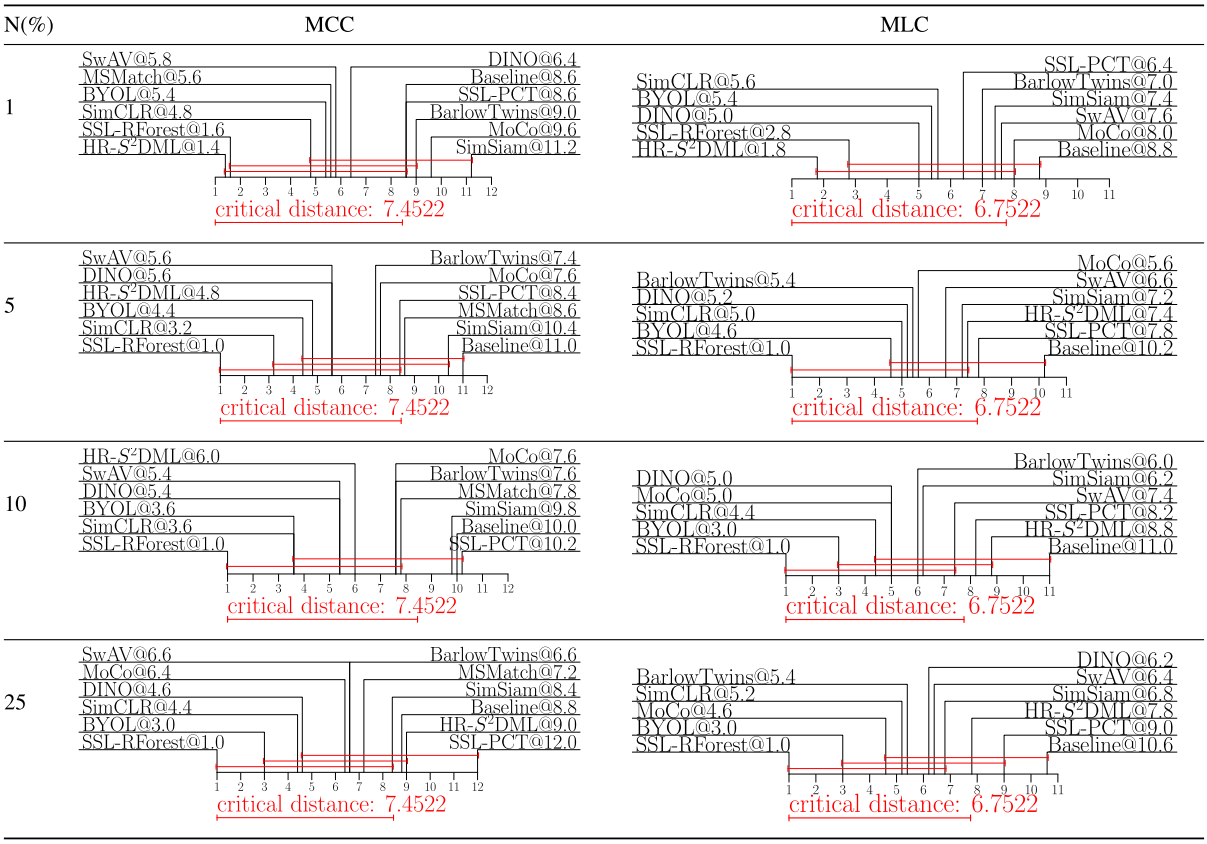



Fig. 9. Statistical analysis of the relative predictive performance of our proposed methods and state-of-the-art methods in terms of AUPRC for MCC datasets and AUPRC for MLC datasets considering the EfficientNet-B2 network architecture. The results are presented in the form of average rank diagrams, one for each percentage of labeled data. In the diagrams, the best ranking methods are positioned at the leftmost side, indicating their superior performance. The significance level is set at 0.05. If the methods are connected with a red line, the differences in performance among them are not statistically significant.

| Samples | Ground Truth | Baseline | CAE | HR-$S^2$DML | SSL-PCT | SSL-RForest |
|---|---|---|---|---|---|---|
|  | bare-soil, pavement, trees, water | bare-soil, pavement, trees, water | bare-soil, pavement, trees, water | bare-soil, pavement, trees, water, grass | bare-soil, pavement, trees, water, grass | bare-soil, pavement, trees, water, grass |
|  | bare-soil, buildings, cars, pavement, trees | bare-soil, buildings, cars, pavement, trees | bare-soil, buildings, cars, pavement, trees | bare-soil, buildings, cars, pavement, trees | bare-soil, buildings, cars, pavement, trees, grass | bare-soil, buildings, cars, pavement, trees |
|  | airplane, grass, pavement | airplane, grass pavement, cars | airplane, grass pavement, cars | airplane, grass pavement, cars | airplane, grass pavement, cars | airplane, grass, pavement |

(a)

| Samples | Ground Truth | Baseline | CAE | HR-$S^2$DML | SSL-PCT | SSL-RForest |
|---|---|---|---|---|---|---|
|  | impervious, water, clutter | impervious, water, clutter vegetation | impervious, water, clutter | impervious, water, clutter | impervious, water, clutter | impervious, water, clutter |
|  | impervious, vegetation, building, car | impervious, vegetation, building, car | impervious, vegetation, building, car | impervious, vegetation, building, car | impervious, vegetation, building, car | impervious, vegetation, building, car |
|  | impervious, clutter | impervious, clutter, car | impervious, clutter, car | impervious, clutter | impervious, clutter | impervious, clutter |

(b)

Fig. 10. Qualitative results of different methods based on 10% of labeled data on: (a) UCM dataset and (b) DFC-15 dataset for MLC of RSIs. The predictions shown in red indicate false positives. Cases shown in blue are false negatives (should have been predicted, but were not).

To determine the statistical significance of the performance differences among the SSL methods across the RSI datasets, we employed the Friedman test and Nemenyi posthoc analysis [70] with the significance level set at $\alpha = 0.05$. The results are presented using average rank diagrams in Fig. 9, where the best-performing methods are positioned at the leftmost side of the diagram (with a rank of 1). Methods with performances that do not differ significantly are connected

with a red line. As depicted in the diagrams, the SSL-RForest is a clear best performer for all percentages of labeled data, except for 1%, where HR-S$^2$DML is ranked the best. The HR-S$^2$DML method loses its advantage as the amount of labeled data increases, with SimCLR and BYOL becoming the closest competitors to SSL-RForest. For $>1\%$ of labeled data, SSL-RForest statistically significantly outperforms the supervised end-to-end baseline.

## V. Conclusion

In this study, we have proposed a framework for semi-supervised multi-label and MCC of RSIs. It consists of two main steps: feature representation learning using a backbone CNN model, to capture the semantically rich content of RSI, followed by semi-supervised PCTs and ensembles.

We have conducted an extensive experiment evaluation covering various aspects of the proposed approach. We use different RSI datasets for MLC and MCC. First, we evaluate the performance of the methods with different amounts of labeled data. Next, we analyze the dependence of their predictive performance on the different network architectures used as backbone feature extractors. Finally, we rigorously evaluate our method against several state-of-the-art self-supervised and SSL methods from the literature.

The results of our experiments demonstrate that the proposed semi-supervised approaches, SSL-PCT and SSL-RForest, clearly benefit from unlabeled data as they consistently outperform their supervised counterparts across different datasets, network architectures, and amounts of labeled data. This was not the case for the semi-supervised end-to-end CAE method, which exhibited improvement over its supervised learning counterpart, only for the EfficientNet-B2 network architecture. Next, the SSL-RForest method emerged as a clear winner among nine state-of-the-art self-supervised and semi-supervised deep learning methods (including BYOL [57], SimSiam [58], SimCLR [59], BarlowTwins [60], MoCo [61], DINO [62], SwAV [63], HR-S$^2$DML [35], and MSMatch [23]). The simpler method, SSL-PCT, even if it cannot compete with the other methods in terms of predictive performance, efficiently builds interpretable models, i.e., classification trees (the only such method among the ones considered).

The findings of this study imply that our semi-supervised framework can effectively reduce the reliance on extensive supervision in RSI multiclass and MLC. This is particularly valuable in real-world scenarios where the cost of labeling is high. Building upon these promising results, future extensions of this work will focus on exploring the incorporation of hierarchical label information to further enhance the predictive performance of semi-supervised MLC of RSIs.

## References


[1] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "BigearthNet: A large-scale benchmark archive for remote sensing image understanding," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2019, pp. 5901–5904.

[2] D. Hou, S. Wang, X. Tian, and H. Xing, "PCLUDA: A pseudo-label consistency learning-based unsupervised domain adaptation method for cross-domain optical remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5600314.

[3] Y. Feng, J. Jiang, H. Xu, and J. Zheng, "Change detection on remote sensing images using dual-branch multilevel intertemporal network," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4401015.

[4] J.-X. Wang, S. Chen, C. H. Q. Ding, J. Tang, and B. Luo, "Semi-supervised semantic segmentation of remote sensing images with iterative contrastive network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[5] Y. Du, L. Du, Y. Guo, and Y. Shi, "Semisupervised SAR ship detection network via scene characteristic learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5201517.

[6] I. Dimitrovski, I. Kitanovski, D. Kocev, and N. Simidjievski, "Current trends in deep learning for Earth observation: An open-source benchmark arena for image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 197, pp. 18–35, Mar. 2023.

[7] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 4, pp. 213–247, Dec. 2022.

[8] D. Hong et al., "Cross-city matters: A multimodal remote sensing benchmark dataset for cross-city semantic segmentation using high-resolution domain adaptation networks," *Remote Sens. Environ.*, vol. 299, Dec. 2023, Art. no. 113856.

[9] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, Aug. 2024.

[10] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, and M. Ryo, "Ten deep learning techniques to address small data problems with remote sensing," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 125, Dec. 2023, Art. no. 103569.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[12] M. Stoimchev, D. Kocev, and S. Džeroski, "Deep network architectures as feature extractors for multi-label classification of remote sensing images," *Remote Sens.*, vol. 15, no. 2, p. 538, Jan. 2023.

[13] Y. Huang et al., "Two-branch attention adversarial domain adaptation network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5540813.

[14] S. Chen and W. Guo, "Auto-encoders in deep learning—A review with new perspectives," *Mathematics*, vol. 11, no. 8, p. 1777, Apr. 2023.

[15] J. Xiao, Y. Bai, A. Yuille, and Z. Zhou, "Delving into masked autoencoders for multi-label thorax disease classification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 3577–3589.

[16] P. Ma, Z. He, W. Ran, and H. Lu, "A transferable generative framework for multi-label zero-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3409–3423, May 2024.

[17] Q. Liu, J. Peng, G. Zhang, W. Sun, and Q. Du, "Deep contrastive learning network for small-sample hyperspectral image classification," *J. Remote Sens.*, vol. 3, p. 25, Jan. 2023.

[18] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," 2020, *arXiv:2001.07685*.

[19] B. Chen, P. Li, X. Chen, B. Wang, L. Zhang, and X.-S. Hua, "Dense learning based semi-supervised object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4805–4814.

[20] J. Zhuang, Z. Wang, and Y. Gao, "Semi-supervised video semantic segmentation with inter-frame feature reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3253–3261.

[21] Q. Liu, M. He, Y. Kuang, L. Wu, J. Yue, and L. Fang, "A multi-level label-aware semi-supervised framework for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–12, 2023.

[22] D. Mishra and O. Hadar, "CLSR: Contrastive learning for semi-supervised remote sensing image super-resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[23] P. Gómez and G. Meoni, "MSMatch: Semisupervised multispectral scene classification with few labels," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11643–11654, 2021.

[24] W. Huang, Y. Shi, Z. Xiong, Q. Wang, and X. X. Zhu, "Semi-supervised bidirectional alignment for remote sensing cross-domain scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 195, pp. 192–203, Jan. 2023.

[25] W. Miao, J. Geng, and W. Jiang, "Semi-supervised remote-sensing image scene classification using representation consistency Siamese network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616614.

[26] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "MixMatch: A holistic approach to semi-supervised learning," 2019, *arXiv:1905.02249*.

[27] Y. Wang et al., "FreeMatch: Self-adaptive thresholding for semi-supervised learning," 2023, *arXiv:2205.07246*.

[28] Y. Oh, D.-J. Kim, and I. S. Kweon, "DASO: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 9786–9796.

[29] B. Zhang et al., "FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 18408–18419.

[30] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10687–10698.

[31] I. Nassar, S. Herath, E. Abbasnejad, W. Buntine, and G. Haffari, "All labels are not created equal: Enhancing semi-supervision via label grouping and co-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7237–7246.

[32] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Computer Vision—ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham, Switzerland: Springer, 2018, pp. 142–159.

[33] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "SimMatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14471–14481.

[34] J. Li, C. Xiong, and S. C. Hoi, "Semi-supervised learning with contrastive graph regularization," in *Proc. ICCV*, 2021, pp. 9475–9484.

[35] J. Kang, R. Fernández-Beltrán, Z. Ye, X. Tong, P. Ghamisi, and A. Plaza, "High-rankness regularized semi-supervised deep metric learning for remote sensing imagery," *Remote Sens.*, vol. 12, no. 16, p. 2603, Aug. 2020.

[36] C. Hu, X.-J. Wu, and J. Kittler, "Semi-supervised learning based on GAN with mean and variance feature matching," *IEEE Trans. Cognit. Develop. Syst.*, vol. 11, no. 4, pp. 539–547, Dec. 2019.

[37] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[38] R. Sharma, S. Mashkaria, and S. P. Awate, "A semi-supervised generalized VAE framework for abnormality detection using one-class classification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1302–1310.

[39] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1144–1158, Feb. 2018.

[40] T. Üstünkök and M. Karakaya, "SS-MLA: A semisupervised method for multi-label annotation of remotely sensed images," *J. Appl. Remote Sens.*, vol. 15, no. 3, Aug. 2021, Art. no. 036509.

[41] J. Levatić, M. Ceci, D. Kocev, and S. Džeroski, "Semi-supervised classification trees," *J. Intell. Inf. Syst.*, vol. 49, no. 3, pp. 461–486, Dec. 2017.

[42] J. Levatić, M. Ceci, D. Kocev, and S. Džeroski, "Semi-supervised predictive clustering trees for (hierarchical) multi-label classification," *Int. J. Intell. Syst.*, vol. 2024, Apr. 2024, Art. no. e5610291. [Online]. Available: https://www.hindawi.com/journals/ijis/2024/5610291/

[43] Y.-F. Li and D.-M. Liang, "Safe semi-supervised learning: A brief introduction," *Frontiers Comput. Sci.*, vol. 13, no. 4, pp. 669–676, Aug. 2019.

[44] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019.

[45] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[46] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[47] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[48] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2010, pp. 270–279.

[49] Y. Hua, L. Mou, and X. X. Zhu, "Relation network for multilabel aerial image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4558–4572, Jul. 2020.

[50] F. Ömrüuzun, B. Demir, L. Bruzzone, and Y. Y. Çetin, "Content based hyperspectral image retrieval using bag of endmembers image descriptors," in *Proc. 8th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Aug. 2016, pp. 1–4.

[51] Y. Hua, L. Mou, and X. X. Zhu, "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 149, pp. 188–199, Mar. 2019.

[52] X. Qi et al., "MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding," *ISPRS J. Photogramm. Remote Sens.*, vol. 169, pp. 337–350, Nov. 2020.

[53] X.-Z. Wu and Z.-H. Zhou, "A unified view of multi-label performance measures," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3780–3788.

[54] F. Pedregosa, S. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Dec. 2011.

[55] S. Cheng, H. Chen, P. Yao, and L. Song, "MLAE: A pretraining method for automatic identification of urban public space," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[56] C. Qiu, A. Yu, X. Yi, N. Guan, D. Shi, and X. Tong, "Open self-supervised features for remote-sensing image scene classification using very few samples," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[57] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[58] X. Chen and K. He, "Exploring simple Siamese representation learning," 2020, *arXiv:2011.10566*.

[59] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*.

[60] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021, *arXiv:2103.03230*.

[61] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019, *arXiv:1911.05722*.

[62] M. Caron et al., "Emerging properties in self-supervised vision transformers," 2021, *arXiv:2104.14294*.

[63] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2020, *arXiv:2006.09882*.

[64] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "$S^4$L: Self-supervised semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1476–1485.

[65] (2023). *PyTorch*. Accessed: Jun. 1, 2023. [Online]. Available: https://pytorch.org/

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[68] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[69] M. Petković, J. Levatić, D. Kocev, M. Breskvar, and S. Džeroski, "CLUSplus: A decision tree-based framework for predicting structured outputs," *SoftwareX*, vol. 24, Dec. 2023, Art. no. 101526.

[70] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.



**Marjan Stoimchev** received the M.S. degree from the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2021. He is currently pursuing the Ph.D. degree with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana.

His main research interests include computer vision, specifically in deep learning, classical machine learning, the combination of both, and multilabel classification of remote sensing images with supervised and semisupervised learning.



**Jurica Levatić** received the Ph.D. degree in semisupervised learning for structured output prediction from the Jožef Stefan Institute Postgraduate School, Ljubljana, Slovenia, in 2017.

From 2018 to 2021, he was a Marie Skłodowska-Curie Post-Doctoral Fellow with the Institute for Research in Biomedicine, Barcelona, Spain. He is currently a Post-Doctoral Researcher with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana. His research interests include cancer biology semisupervised learning and the application of machine learning to solve practically relevant problems.



**Dragi Kocev** received the Ph.D. degree in learning ensemble models for predicting structured outputs from the Jožef Stefan Institute Postgraduate School, Ljubljana, Slovenia, in 2011.

He was a Visiting Research Fellow at the University of Bari, Bari, Italy, from 2014 to 2015. He is currently a Researcher with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana. He was involved with the EU-funded projects TRUSTroke, IQ, PHAGOSYS, and HBP. He was a Co-Coordinator of the FP7 FET Open Project MAESTRA and a Coordinator of ESA-funded activities AiTLAS and GalaxAI.

Dr. Kocev is a member of the PC for several conferences (e.g., DS, ECML PKDD, AAAI, IJCAI, and KDD). He served as the PC Co-Chair for DS 2014 and the Journal Track Co-Chair for ECML PKDD 2017. He is an Action Editor of *Expert Systems with Applications*, *Machine Learning*, and *Data Mining and Knowledge Discovery*.



**Sašo Džeroski** is currently the Head of the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia. His department develops artificial intelligence methods for machine learning and decision support and uses them to solve practical problems from agriculture and environmental sciences, medicine and life sciences, and space operations/Earth observation. His research focuses on artificial intelligence for science, including machine learning from complex data and in the presence of domain knowledge, learning models of dynamical systems, and developing ontologies for describing different branches of computer science research.

Dr. Džeroski is a member of the Macedonian Academy of Sciences and Arts and Academia Europea (European Academy of Humanities, Letters and Sciences). He is a fellow of EurAI, European Association of Artificial Intelligence, awarded in recognition for his “Pioneering Work in AI and Outstanding Service for the European AI Community.” He is currently the Vice President of Slovenian Artificial Intelligence Society.

## 4.2 Dimensionality Reduction for Efficient Semi-Supervised Learning From Remote Sensing Images

While the results confirmed the effectiveness of SSL with PCTs and ensembles, we observed that the practical application of these methods was limited by computational overhead when using high-dimensional CNN features. To address this, the framework was extended with PCA-based dimensionality reduction to improve efficiency while maintaining predictive performance.

We explored reduction levels corresponding to 60%, 80%, and 95% explained variance and evaluated their impact on both predictive accuracy and runtime. The results show that SSL methods preserve their superiority over supervised baselines even after dimensionality reduction. SSL-PCTs consistently improve predictive performance, while SSL-RForests maintain most of their advantages. Moreover, dimensionality reduction leads to dramatic speed-ups in induction time, often by two to three orders of magnitude, while in some cases even improving predictive accuracy compared to the original feature space.

These findings demonstrate that PCA can remove redundant or noisy components from CNN features, making SSL with PCTs and ensembles not only more efficient but sometimes even more accurate. EfficientNet-based features continue to outperform those derived from ResNet across all reduction levels.

The paper included in this section is:

- M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Dimensionality Reduction for Efficient Semi-supervised Learning from Remote Sensing Images", in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Revised Selected Papers, Part III*, *Communications in Computer and Information Science*, vol. 2560, in press, Cham: Springer, 2026.

**The contributions of Marjan Stoimchev to this paper are as follows.** Marjan Stoimchev extended the SSL framework with PCA-based dimensionality reduction, conducted efficiency-focused experiments, analyzed the computational trade-offs, and prepared and revised the manuscript.

**Code availability.** All code developed for the experiments presented in this section, including the evaluation scripts of the SSL-PCTs and SSL-RForest extended with PCA-based dimensionality reduction for CNN-derived features, is publicly available at: `https://github.com/marjanstoimchev/pcts_dimensionality_reduction`.

# Dimensionality Reduction for Efficient Semi-supervised Learning from Remote Sensing Images

Sintija Stevanoska[1,2] *, Marjan Stoimchev[1,2] *, Jurica Levatić[1], and Sašo Džeroski[1,2]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia
{sintija.stevanoska, marjan.stoimchev}@ijs.si

**Abstract.** While large quantities of image data are available, labels assigned to images are scarce, especially in land-cover/land-use classification of remotely-sensed images (RSI). Semi-supervised learning (SSL), which can exploit both labeled and unlabeled images, is thus becoming increasingly important in the classification of RSI. A recent approach successfully uses (ensembles of) predictive clustering trees (PCTs) for SSL in RSI classification, in conjunction with deep neural networks (DNNs) as feature extractors. The large number of features extracted by DNNs in this context raises serious computational complexity issues, as features are used as both inputs and targets by SSL PCTs.
We address this problem by applying dimensionality reduction to the space of features extracted by the DNNs. When learning (ensembles) of PCTs for SSL classification, the reduced space is used as input instead of the original features. This greatly reduces the time complexity of SSL, while preserving the predictive performance. We verify that this holds for both PCTs and ensembles thereof, for different DNN architectures, on several multi-class classification datasets of RSI.

**Keywords:** Remote sensing · Dimensionality reduction · Semi-supervised learning · Multi-class classification.

## 1 Introduction

The exponential growth in the generation of image data, including remote sensing images (RSI), has fueled significant advancements in the field of computer vision. Remote sensing provides invaluable insights into our environment, allowing extraction of vital information on topics such as land cover, vegetation health, and climate patterns. Consequently, the remote sensing community has directed considerable attention and research efforts towards image classification [7].

* These authors contributed equally to this work

Multi-class classification (MCC) involves categorizing each RSI into one of multiple distinct classes. Machine learning (ML) techniques, particularly deep neural networks (DNN) [7] and convolutional neural networks (CNN) [10], have demonstrated their efficacy in this domain. In recent years, transfer learning and fine-tuning have further improved RSI classification [9]. Transfer learning leverages knowledge from pre-trained models on large-scale datasets to enhance the performance on smaller-scale RSI datasets, while fine-tuning further refines the pre-trained models, adapting them to the specific characteristics of the RSI data.

Supervised learning, which relies on labeled samples, has traditionally been employed for RSI classification. However, the availability of labeled samples may be limited or they may be costly to acquire. In such cases, semi-supervised learning (SSL) offers a compelling alternative, as it combines labeled and unlabeled samples to enhance classification performance [14], [6]. SSL thus uses many more examples than supervised learning and also relies on the (high dimensional) descriptive space values of unlabeled examples to build the predictive models. This leads to increased induction times, which can render SSL unusable on remote sensing datasets that are often large and highly dimensional.

In this work, we address this issue by proposing a framework that integrates deep learning, dimensionality reduction, and classical ML algorithms for efficient SSL from RSI. We use a CNN-based feature extractor to embed the rich semantic content of RSI into a set of descriptive features [11], that are then used as input for semi-supervised predictive clustering trees (PCTs) and random forests [6]. To deal with the computational complexity of SSL algorithms on the large number of features extracted by CNNs, we rely on Principal Component Analysis (PCA) to identify the most informative and uncorrelated features, transforming the high-dimensional original feature space into a new, lower-dimensional, set of orthogonal components. We explore whether we can reduce the time complexity of SSL methods via dimensionality reduction while maintaining the advantage in the predictive performance of SSL methods over supervised ones.

We evaluate the effectiveness of our framework through extensive experiments conducted on five multi-class classification RSI datasets. The results demonstrate that our approach significantly reduces the induction time of the semi-supervised PCTs and random forests while preserving their classification performance.

## 2 Methods

### 2.1 End-to-end learning and feature extraction

The first component of our methodology is the feature extraction module, which uses fully convolutional backbone feature extractors to encode the distinctive characteristics of an image in a lower-dimensional space, as described by Stoimchev et al. [11]. Using these techniques, the system efficiently captures and represents the essential information within the images, enabling effective execution of the subsequent processing and analysis stages. We used two different families of CNN models, namely, ResNets and EfficientNets.

ResNets are a group of deep CNN architectures that apply residual learning to facilitate the training of deep networks [4], effectively addressing vanishing gradients. ResNet follows the $3 \times 3$ convolutional layer design introduced by the VGG architecture [5]. By adjusting the number of channels and residual blocks, various ResNet models can be created, such as the deeper ResNet-152 with 152 layers used in this work.

EfficientNets do not suffer from over-parameterization or arbitrary scaling of network dimensions like width, depth, and resolution. Instead, they employ a consistent scaling approach using predetermined coefficients that uniformly scale each dimension [12].

The backbone models are initialized with weights learned on ImageNet [3]. They are further fine-tuned on each of the small datasets of RSI for MCC. We optimize the model parameters using the cross-entropy loss, and the softmax activation to convert the logit vectors into probabilistic scores. Lastly, the $d$-dimensional feature representations are extracted from the average pooling layer from the respective backbone CNN models.

### 2.2 Dimensionality reduction

For the dimensionality reduction, we use PCA [13] as a simple, widely adopted method that identifies orthogonal axes, referred to as *principal components*, which capture the maximum variance in the data. PCA projects the data onto a lower-dimensional subspace, effectively reducing dimensionality while preserving essential information.

In the context of semi-supervised MCC for RSI, this study uses PCA for efficient data representation. The feature extraction process, outlined in Section 2.1, results in datasets with high numbers of features, which can negatively impact the performance of the semi-supervised algorithms. Thus, the primary objective is to enhance efficiency while maintaining satisfactory performance and complexity levels. We explore three PCA configurations, with explained variance thresholds set to 60%, 80%, and 95%, resulting in increasingly different reduced feature sets.

### 2.3 SSL of PCTs

***Predictive clustering trees.*** The PCT framework conceptualizes a decision tree as a hierarchy of clusters, where the top node represents a cluster containing all the data, which is recursively divided into smaller clusters as we move down the tree. The `CLUS` system [8], available at `https://github.com/knowledge-technologies/clus`, implements the PCT framework, including PCTs and ensembles thereof. PCTs are constructed using the standard top-down induction of decision trees (TDIDT) algorithm. Test selection for internal nodes is driven by impurity reduction — variance for regression and Gini impurity for classification.

The key distinction between learning PCTs and standard decision trees lies in the ability of PCTs to instantiate the impurity function (evaluating the quality of

splits) and the prototype function (assigning labels to each leaf) as parameters. This allows PCTs to handle a variety of predictive modelling tasks, including MCC, multi-label classification, and multi-target regression.

***PCTs for SSL.*** In SSL, the data consists of a descriptive space $X$ and a target space $Y$, with a mix of labeled examples $E_l$ and unlabeled examples $E_u$. Two key modifications allow PCTs to extend to the SSL scenario [6]: First, both labeled and unlabeled examples are used during model training. Second, the variance function used for determining optimal splits is modified to use both $X$ and $Y$:

$$Var_f(E) = w \cdot Var_f^Y(E) + (1 - w) \cdot Var_f^X(E) \quad (1)$$

The weight parameter $w$ controls the influence of $X$ versus $Y$ in the split-impurity calculation, controlling the level of supervision. A $w$ close to 1 emphasizes supervised learning, while $w$ close to 0 favors unsupervised learning. Splits maximizing variance reduction are selected.

The above semi-supervised approach can be extended to learning ensembles, i.e., random forests of PCTs, by constructing a set of semi-supervised PCTs, combining the benefits of SSL and ensemble learning with random forests.

## 3 Experimental setup

### 3.1 Datasets

To evaluate the proposed approach, we use five RSI datasets for MCC: OPTIMAL-31 [15], RESISC-45 [2], RSSCN-7 [1], AID [16] and UCM [17]. OPTIMAL-31 is a dataset comprising 31 distinct categories of scene classification, with images collected from Google Earth and sized at $256 \times 256$ pixels. The RESISC-45 dataset is a publicly available benchmark for RSI scene classification, consisting of 31,500 images at $256 \times 256$ resolution and covering 45 scene classes. In contrast, the RSSCN-7 dataset contains 2,800 scene classification images obtained from Google Earth, with a resolution of $400 \times 400$. For AID and UCM, which can also be used for multi-label classification, we used the multi-class versions of the datasets, with 21 and 30 scene classes, respectively.

### 3.2 Evaluation protocol

To assess the effectiveness of the methods, we use train/test splits of the datasets. In the case of the AID dataset, we use its predefined split where the training and test set sizes are approximately 80% and 20% of the total, while for the remaining datasets, we randomly split them into training (80%) and test (20%) parts accordingly. We adopt an iterative stratified sampling strategy to ensure that the relative frequency of the labels in the datasets is preserved to the maximum extent possible. Additionally, we allocate a validation set of 10% from the training data. This validation set is solely used to monitor learning progress and provide an unbiased estimate of the model fit when tuning hyper-parameters.

To investigate the impact of labeled and unlabeled data on the predictive performance, we simulate a semi-supervised scenario by using different percentages of labeled examples, i.e., 1%, 5%, 10%, and 25%, sampled from the combined train and validation sets, while the remaining samples are treated as unlabeled data, i.e., their labels are removed. To provide a more reliable estimate, for each percentage of labeled data, we repeat the experiments five times using different random initializations (and report averages). The final predictive performance is assessed on the fixed-sized test set, where the labels of the examples are provided.

### 3.3 Parameter settings and training details

We used two different network architectures to extract features: ResNet-152 and EfficientNet-B2. Starting with weights pre-trained on ImageNet, they are trained on the train set for 25 epochs on an NVIDIA A100 (40GB) GPU. Their hyperparameters are optimized with the Adam optimizer, where the learning rate of $1 \times 10^{-4}$ remains fixed during the training process and the size of the mini-batches is 128. After the calibration process, the feature extraction step is applied to the different subsets of images (train, test) for a particular dataset. The backbone models remain frozen during this feature extraction process to avoid further gradient updates, resulting in $d$-dimensional feature vectors.

We apply PCA to the feature vectors to construct feature sets with varying numbers of principal components that cumulatively explain 60%, 80%, and 95% of the variance. We conduct a comparative analysis of the impact that the varying levels of dimensionality reduction have on the predictive performance and efficiency of SSL with PCTs. The summary of the dimensions, i.e., the number of features, before and after the application of PCA is given in Table 1.

The original and reduced feature representations are used as inputs to the PCTs and the random forest. Single supervised and SSL PCTs are pruned using `M5` pruning, while for random forest we use 100 unpruned trees and a feature subset size equal to the square root of the total number of features. Furthermore, for the SSL version of the methods, we use 3-fold internal cross-validation on the training part of the dataset to optimize the $w$ parameter defined in Eq. 1. We optimize the $w$ parameter within the range of 0 to 1, with increments of 0.1.

We compare the performance of semi-supervised PCTs and random forests (RForest) with different numbers of principal components used to calculate the variance. These are denoted in the naming of methods, e.g., SSL-60 denotes that the semi-supervised method was used with PCA components that explain 60% of variance of the original features, while SSL-100 denotes that all original features were used to calculate the variance $Var_f$ in `CLUS` (i.e., '-100' denotes that dimensionality reduction was not performed). We also compare to the baseline of supervised learning (SL), which does not use the input features (or principal components) in the calculation of $Var_f$.

Table 1: The dimensions of the feature spaces before and after the use of PCA. EV is the cumulative explained variance (in %).

| | | | EV(%) | | |
|---|---|---|---|---|---|
| Dataset | Features extracted | Original | 95 | 80 | 60 |
| OPTIMAL-31 | ResNet-152 | 2048 | 48 | 21 | 12 |
| UCM | | 2048 | 159 | 19 | 12 |
| RSSCN7 | | 2048 | 163 | 12 | 5 |
| AID | | 2048 | 54 | 20 | 12 |
| RESISC45 | | 2048 | 61 | 26 | 15 |
| OPTIMAL-31 | EfficientNet-B2 | 1408 | 192 | 27 | 16 |
| UCM | | 1408 | 36 | 15 | 10 |
| RSSCN7 | | 1408 | 30 | 6 | 4 |
| AID | | 1408 | 214 | 27 | 16 |
| RESISC45 | | 1408 | 416 | 47 | 24 |

### 3.4 Performance metrics

To assess the performance of the ML algorithms on the classification tasks, the area under the precision-recall curve (AUPRC) was computed and averaged over all classes, providing an overall measure of the algorithms' performance in handling the multi-class classification problem. We also measured the induction time of the methods to evaluate and compare the runtime of the algorithms, considering the size and complexity of the input data. To ensure reliable and accurate measurements, all the experiments were conducted on the same computing platform (`AMD EPYC 7702 2GHz` 64 CPUs and 1 TB of RAM), and the execution times were averaged over multiple runs to mitigate potential variations.

## 4 Results

We first examine the impact of unlabeled data on the predictive performance of SSL, both with and without feature dimensionality reduction. We next investigate the effect of dimensionality reduction on the induction times of the methods. Lastly, we investigate the influence of the different types of feature extractors on the predictive performance.

### 4.1 Performance Curves

Table 2 presents the predictive performance of the algorithms with different percentages of labeled data and varying levels of dimensionality reduction. First, we can observe that SSL PCTs and random forests (without dimensionality reduction) consistently improve performance over their supervised counterparts across all datasets and different percentages of labeled data. The degree of improvement of SSL methods over supervised methods is larger for SSL PCTs than for semi-supervised random forests.

Next, we investigate the effect of dimensionality reduction on the predictive performance of the SSL algorithms. We can observe (Table 2) that in the case of PCTs, the superiority of SSL is preserved when dimensionality reduction is applied. The degree of improvement depends on the level of the dimensionality reduction. Using the principal components explaining 95% or 80% of variance

Table 2: Comparison of the predictive performance ($\overline{\text{AUPRC}}$) of supervised and semi-supervised PCTs and random forests (RForest) for different percentages of labeled data (1-25%), and different levels of dimensionality reduction (60-95% of variance explained). We only show the predictive performance for the better network architecture used as feature extractor (EfficientNet-B2). The setting with the original features (i.e., no dimensionality reduction) is denoted with SSL-100. The best performance for each dataset is shown in bold.

| | PCTs | | | | | RForest | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1(%) | | | | | 1(%) | | | | |
| Dataset | SSL-60 | SSL-80 | SSL-95 | SSL-100 | SL | SSL-60 | SSL-80 | SSL-95 | SSL-100 | SL |
| OPTIMAL-31 | 0.115 | 0.158 | **0.181** | 0.119 | 0.048 | 0.386 | **0.398** | **0.398** | 0.383 | 0.327 |
| UCM | 0.348 | 0.329 | **0.335** | 0.251 | 0.085 | 0.607 | 0.616 | **0.618** | 0.593 | 0.539 |
| RSSCN7 | 0.815 | 0.838 | **0.901** | 0.810 | 0.342 | 0.973 | **0.980** | 0.975 | 0.979 | 0.960 |
| AID | 0.222 | **0.285** | 0.246 | 0.211 | 0.076 | 0.526 | 0.524 | 0.528 | **0.531** | 0.455 |
| RESISC45 | 0.687 | **0.722** | 0.683 | 0.634 | 0.290 | 0.981 | 0.984 | 0.983 | **0.985** | 0.970 |
| Rank | 2.600 | **1.800** | **1.800** | 3.800 | 5.000 | 3.400 | **2.000** | **2.000** | 2.400 | 5.000 |
| | 5(%) | | | | | 5(%) | | | | |
| OPTIMAL-31 | 0.596 | **0.672** | 0.662 | 0.573 | 0.180 | 0.884 | 0.890 | **0.893** | 0.877 | 0.841 |
| UCM | 0.831 | **0.857** | 0.849 | 0.834 | 0.377 | 0.994 | 0.996 | **0.997** | 0.997 | 0.986 |
| RSSCN7 | 0.865 | 0.884 | **0.903** | 0.820 | 0.726 | 0.979 | **0.985** | **0.985** | 0.983 | 0.981 |
| AID | 0.682 | **0.684** | 0.639 | 0.620 | 0.282 | **0.939** | 0.937 | 0.938 | 0.931 | 0.908 |
| RESISC45 | **0.791** | 0.790 | 0.756 | 0.752 | 0.605 | 0.986 | 0.987 | 0.988 | **0.989** | 0.987 |
| Rank | 2.600 | **1.400** | 2.200 | 3.800 | 5.000 | 3.600 | 2.400 | **1.400** | 2.600 | 4.400 |
| | 10(%) | | | | | 10%) | | | | |
| OPTIMAL-31 | 0.720 | 0.764 | **0.769** | 0.705 | 0.409 | 0.964 | **0.966** | 0.964 | 0.962 | 0.948 |
| UCM | 0.837 | **0.872** | 0.864 | 0.849 | 0.584 | 0.996 | 0.996 | 0.997 | **0.998** | 0.997 |
| RSSCN7 | 0.869 | 0.890 | **0.904** | 0.835 | 0.770 | 0.980 | **0.986** | **0.986** | 0.984 | 0.984 |
| AID | **0.732** | 0.719 | 0.681 | 0.691 | 0.484 | 0.980 | 0.976 | 0.977 | **0.988** | 0.979 |
| RESISC45 | 0.817 | **0.818** | 0.784 | 0.783 | 0.682 | 0.986 | 0.988 | 0.988 | **0.989** | 0.988 |
| Rank | 2.600 | **1.600** | 2.200 | 3.600 | 5.000 | 3.600 | 2.600 | 2.200 | **2.000** | 3.000 |
| | 25(%) | | | | | 25(%) | | | | |
| OPTIMAL-31 | 0.750 | **0.783** | 0.777 | 0.728 | 0.616 | 0.973 | 0.975 | 0.975 | **0.976** | **0.976** |
| UCM | 0.856 | **0.872** | 0.866 | 0.844 | 0.789 | 0.996 | 0.997 | **0.998** | **0.998** | **0.998** |
| RSSCN7 | 0.887 | 0.900 | **0.910** | 0.843 | 0.836 | 0.982 | 0.986 | **0.987** | 0.985 | 0.983 |
| AID | **0.735** | 0.719 | 0.699 | 0.702 | 0.615 | 0.980 | 0.979 | 0.976 | **0.988** | 0.986 |
| RESISC45 | 0.839 | **0.841** | 0.816 | 0.804 | 0.735 | 0.987 | 0.989 | 0.989 | **0.990** | 0.989 |
| Rank | 2.400 | **1.400** | 2.400 | 3.800 | 5.000 | 4.600 | 3.000 | 2.400 | **1.400** | 2.000 |
| Sum of ranks | 10.200 | **6.200** | 8.600 | 15.000 | 20.000 | 15.200 | 10.000 | **8.000** | 8.400 | 14.400 |

yielded the best predictive performance in most of the datasets (Table 2), outperforming even the use of the original features.

In the case of semi-supervised random forests, there is some loss of predictive performance after the application of PCA. Still, with 95% of variance explained, the superiority of semi-supervised random forests over supervised ones is largely preserved, while for smaller percentages of variance explained, semi-supervised random forests tend to lose their advantage in the predictive performance.

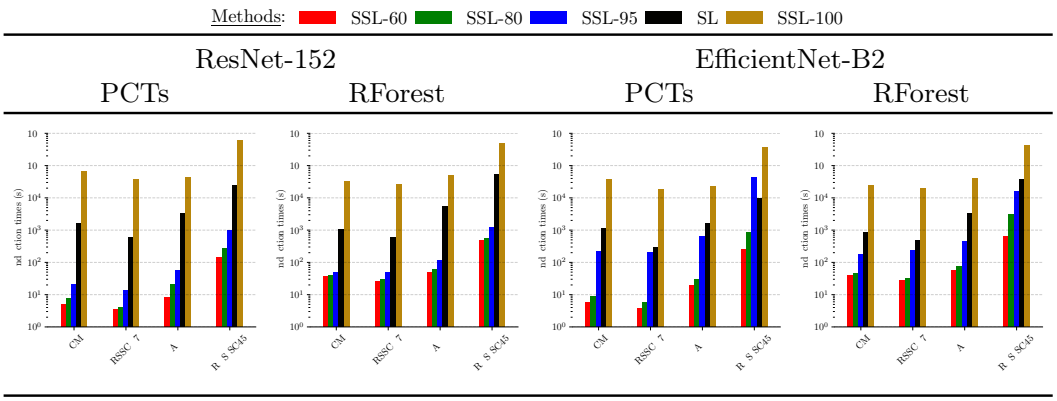



Fig. 1: Induction times in seconds for semi-supervised and supervised learning algorithms for 25% labeled data and different levels of dimensionality reduction.

## 4.2 Induction times

Semi-supervised PCTs and random forests compute the impurity of the splits in the tree considering both the descriptive and the target spaces. This can lead to a significant increase in the induction times for datasets with a large number of descriptive features, compromising the practical utility of these algorithms.

We observe that dimensionality reduction drastically decreases the induction times for semi-supervised algorithms (Fig. 1): We observe 2 to 3 orders of magnitude of speed-up as compared to SSL with the original features (SSL-100). SSL PCTs and random forests built with principal components are faster to learn even than their supervised counterparts. The principal component features with 95% of variance explained were learned 67 times faster (on average across datasets) than SSL PCTs with the features extracted by the EfficientNet-B2 architecture and 1677 times faster on the features from the ResNet-152 architecture. Similarly, it yielded 81 and 438 times speed-up for SSL random forests on the EfficientNet-B2 and ResNet-152 feature sets respectively. The ResNet-152 architecture has more features (Table 1), hence the speed-up is more notable. Note that, for brevity, in Fig. 1 we present the induction times only for 25% of labeled examples. The conclusions for other amounts of labeled data are analogous.

## 4.3 Comparison of Different Network Architectures

In this section, we investigate the influence of the different types of network architectures on the predictive performance of the models. The results (Fig. 2) show that the EfficientNet-B2 architecture slightly outperforms the ResNet-152 architecture. The results are consistent across different amounts of labeled data and different levels of dimensionality reduction for both supervised and semi-supervised PCTs and random forests. ResNet-152 shows better performance (on average) in only a few cases with 1% of labeled data (SSL-95 PCTs, SL PCTs).

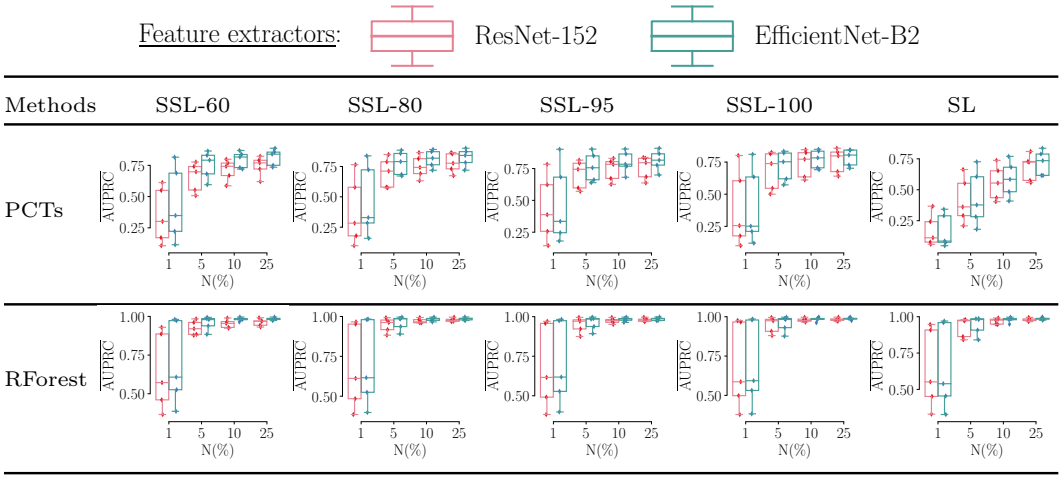



Fig. 2: The distributions of $\overline{\text{AUPRC}}$ for ResNet-152 (red) and EfficientNet-B2 (blue) across different datasets.

## 5 Conclusion

In this work, we explored the possibility of using dimensionality reduction to mitigate the issue of increased induction times of semi-supervised PCTs and random forests in high-dimensional RSI classification data. These algorithms outperform traditional supervised algorithms, but they can be difficult to apply to highly-dimensional remote sensing datasets due to the high-induction time that stems from using a high number of unlabeled images and relying on both descriptive features and the target variable. We proposed a framework that includes dimensionality reduction based on PCA and showed that the superiority in the predictive performance of semi-supervised PCTs and random forests over their supervised counterparts is preserved while the induction time is drastically reduced. We observed up to 2 orders of magnitude speed-up for the EfficientNet-B2 features and up to 3 orders for the ResNet-152 features.

One limitation of dimensionality reduction in the case of single trees is that their interpretability is compromised, i.e., the nodes in the trees use the principal components as features. These features can be projected back to the original descriptive space to rescue some of the interpretability, however, such projections would still be aggregates of the original descriptive features, and as such possibly more difficult to interpret for domain experts.

In future work, we aim to explore the effect of dimensionality reduction on the performance of semi-supervised methods in domains other than remote image classification and other predictive tasks, including multi-target regression and (hierarchical) multi-label classification (i.e., MTR, MLC, or HMLC).

**Acknowledgements** We acknowledge the support of the European Commission via the projects ASSAS (grant number 101059682), ELIAS (101120237), INQUIRE (101057499), PARC (101057014), and TAILOR (952215), as well as by the Slovenian Research Agency via the research program P2-0103 and the research projects J1-3033, J2-2505, J2-4452, J2-4660, J3-3070, J4-3095, J5-4575, J7-4636, J7-4637, and N2-0236.

## References

1. Ali, N., Zafar, B.: RSSCN7 image dataset (2018), https://figshare.com/articles/dataset/RSSCN7_Image_dataset/7006946/1
2. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE **105**(10), 1865–1883 (2017)
3. Deng, J., Dong, W., Socher, R., Li, L., Kai, L., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 248–255 (2009)
4. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep Residual Learning for Image Recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778 (2016)
5. Karen, S., Andrew, Z.: Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR), 1–14 (2015)
6. Levatić, J., Ceci, M., Kocev, D., Džeroski, S.: Semi-supervised classification trees. Journal of Intelligent Information Systems **49**, 461–486 (2017)
7. Li, Y., Zhang, H., Xue, X., Jiang, Y., Shen, Q.: Deep learning for remote sensing image classification: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8**, e1264 (2018)
8. Petković, M., Levatić, J., Kocev, D., Breskvar, M., Džeroski, S.: CLUSplus: A decision tree-based framework for predicting structured outputs. SoftwareX **24**, 101526 (2023)
9. Shabbir, A., Ali, N., Ahmed, J., Zafar, B., Rasheed, A., Sajid, M., Ahmed, A., Dar, S.H.: Satellite and scene image classification based on transfer learning and fine tuning of ResNet50. Mathematical Problems in Engineering **2021**, 1–18 (2021)
10. Song, J., Gao, S., Zhu, Y., Ma, C.: A survey of remote sensing image classification based on CNNs. Big Earth Data **3**, 232–254 (2019)
11. Stoimchev, M., Kocev, D., Džeroski, S.: Deep network architectures as feature extractors for multi-label classification of remote sensing images. Remote Sensing **15**, 538 (2023)
12. Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning (ICML). PMLR **97**, 6105–6114 (2019)
13. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analyzers. Neural Computation **11**, 443–482 (1999)
14. Vatsavai, R.R., Shekhar, S., Burk, T.E.: A semi-supervised learning method for remote sensing data mining. IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 5 pp.–211 (2005)
15. Wang, Q., Liu, S., Chanussot, J., Li, X.: Scene classification with recurrent attention of VHR remote sensing images. IEEE Transactions on Geoscience and Remote Sensing **57**, 1155–1167 (2018)
16. Xia, G.S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Lu, X., Zhang, L.: AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. IEEE Transactions on Geoscience and Remote Sensing **55**, 3965 – 3981 (2017)
17. Yang, Y., Newsam, S.: Bag-of-visual-words and spatial extensions for land-use classification. ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, 270–279 (2010)

# Chapter 5

# SSL-MAE: Adaptive Semi-Supervised Learning Framework for Multi-Label Classification of Remote Sensing Images Using Masked Autoencoders

This chapter presents our contributions concerning end-to-end SSL for RSI classification. We provided a general overview of SSL in Section 1.1 and described both MCC and MLC tasks in Section 2.1. Here, we present SSL-MAE, an adaptive framework that unifies self-supervised representation learning with discriminative classification to avoid the limitations of traditional two-stage SSL approaches. The motivation stems from the computational inefficiency and suboptimal feature utilization inherent in separate pre-training and fine-tuning phases, where learned representations may not be optimally aligned with the downstream classification objectives. The main contributions are the following:

1. We propose a novel end-to-end SSL multi-label classification learning method (SSL-MAE) that combines the strengths of discriminative learning and self-supervised learning while avoiding the need for two-step learning processes.

2. We design a joint loss function that combines supervised and self-supervised loss terms into a single learning objective, enabling SSL and adaptation of the supervision amount to the dataset at hand through both grid search optimization and learnable weight mechanisms.

3. We demonstrate that the proposed adaptive joint learning can be integrated with other state-of-the-art self-supervised methods (BYOL and MoCo), converting them into SSL end-to-end solutions and significantly enhancing their predictive performance, particularly for scenarios with limited labeled data.

4. We perform rigorous evaluation demonstrating that SSL-MAE successfully exploits unlabeled data and consistently outperforms supervised baselines across all datasets and both learning tasks (MCC and MLC). We show that SSL-MAE outperforms competing state-of-the-art SSL methods while achieving competitive performance against self-supervised methods, particularly excelling in low-data regimes.

5. We provide comprehensive qualitative analysis through embedding visualizations and attention maps, revealing how SSL-MAE learns more discriminative feature representations and focuses on semantically relevant image regions compared to purely supervised or unsupervised approaches.

The initial work was motivated by the need to overcome the limitations of two-stage SSL approaches, where separate pre-training and fine-tuning phases often lead to suboptimal feature alignment with downstream tasks and increased computational overhead. To address these challenges, we developed SSL-MAE as an end-to-end solution that simultaneously optimizes self-supervised reconstruction and discriminative classification objectives through adaptive joint learning.

From the hypotheses we defined in Section 1.3, this chapter addresses the following:

**H4** End-to-end semi-supervised learning frameworks for image analysis that unify self-supervised representation learning with task-specific discriminative learning provide better generalization and computational efficiency than traditional two-stage SSL approaches. Moreover, incorporating diverse self-supervised pretext tasks enhances adaptability and performance.

The paper included in this Chapter is:

- Stoimchev, M., Levatić, J., Kocev, D., Džeroski, S. (2025), "SSL-MAE: Adaptive Semisupervised Learning Framework for Multilabel Classification of Remote Sensing Images Using Masked Autoencoders", in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18, 14882-14896.

**The contributions of Marjan Stoimchev to this paper are as follows**. He contributed to the design and implementation of the SSL-MAE framework, including the adaptive joint learning mechanism and its integration with various self-supervised methods. He participated in the comprehensive experimental design across multiple RSI datasets, conducted the extensive evaluations, and performed the qualitative analysis including embedding visualizations and attention map analysis. He drafted the paper and revised it following feedback from the co-authors and the reviewers.

**Code availability.** The implementation of the SSL-MAE model presented in this chapter is publicly available at: `https://github.com/marjanstoimchev/SSL-MAE`.

# SSL-MAE: Adaptive Semisupervised Learning Framework for Multilabel Classification of Remote Sensing Images Using Masked Autoencoders

Marjan Stoimchev, Jurica Levatić, Dragi Kocev, and Sašo Džeroski

***Abstract*—The increasing volume of remotely sensed imagery (RSI) requires efficient processing and extraction of meaningful information. Modern deep learning architectures excel in various tasks but typically require large labeled datasets, which are often scarce in RSI due to the tedious labeling of complex heterogeneous landscapes containing multiple semantic categories. This can limit the potential of supervised deep learning methods. To address this, we propose SSL-MAE, a novel semisupervised learning method based on a masked autoencoder. Our approach unifies self-supervision and discriminative learning within a single, end-to-end framework, leveraging both abundant unlabeled data and limited labeled data. In addition, we introduce an adaptive mechanism to control the level of supervision during learning, crucial for balancing prediction quality with effective use of unlabeled data. To validate SSL-MAE's effectiveness, we conducted comprehensive experiments on ten publicly available RSI datasets (five multiclass and five multilabel classification). The results show that our method outperforms state-of-the-art semisupervised methods and performs favorably against self-supervised methods when labeled data are scarce. Finally, our results suggest that the proposed adaptive joint learning strategy is not restricted to a single design choice; instead, when integrated with different state-of-the-art self-supervised approaches, it significantly enhances their predictive performance.**

***Index Terms*—Masked autoencoders (MAEs), multilabel classification (MLC), remote sensing images, self-supervised learning, semisupervised learning (SSL).**

## I. Introduction

THE assignment of semantic labels to remotely sensed imagery (RSI) has great importance in a wide range of real-world applications, such as environmental monitoring, disaster management, agriculture, urban planning, or biodiversity conservation [1]. Due to the ever increasing volumes of RSI available nowadays, machine learning methods are becoming indispensable for their automatic processing and automatic knowledge extraction. Deep learning (DL) methods proved to be especially effective for these tasks since they can automatically learn meaningful representations of raw input images and yield excellent accuracy in RSI classification [2].

The excellent classification performance of DL methods, however, typically comes with a cost of large labeled training datasets. This poses a critical limitation in the remote sensing (RS) domain, where labeled training examples are often a scarce commodity due to tedious labeling. Namely, the complexity of visual contexts in RS images and high correlations among labels make labeling difficult, especially when predicting composite concepts. In addition, RS imagery often covers vast, remote areas like polar and mountainous regions, where data collection is expensive, dangerous, and logistically challenging. Accurate labeling also requires specialized knowledge and equipment, such as field surveys for soil properties. In addition, RS monitors dynamic phenomena with rapid and unpredictable changes, such as natural disasters and agricultural landscapes [3].

A viable solution to this challenge are semisupervised methods due to their ability to leverage information from both limited labeled data and abundant unlabeled data [4]. A possible alternative are also self-supervised methods, which are intrinsically unsupervised and can be trained on various "pretext tasks" and then fine-tuned with a small amount of labeled data to perform RSI classification [5], [6]. This process can be impractical due to the need for extensive pretraining on large unlabeled datasets and subsequent fine-tuning on limited labeled data, which together demand a significant amount of time and computational resources, making the approach less feasible in many real-world scenarios. Semisupervised methods, on the other hand, *simultaneously* exploit both unlabeled and labeled examples during model training, which avoids two-stage learning (pretraining followed by fine-tuning). This can result in greater generalisability and efficiency of semisupervised learning (SSL) [4].

Recent advancements in SSL have yielded significant success [7], with the most widely used modern SSL techniques tailored to address the task of multiclass classification (MCC) [8], [9], [9], [10]. When applied to the field of RS, MCC translates to assigning exactly *one* of several predefined classes to a satellite image [11], [12], [13]. However, this can lead to an oversimplification of the classification since RS images are

Received 17 January 2025; revised 13 May 2025; accepted 4 June 2025. Date of publication 9 June 2025; date of current version 24 June 2025. This work was supported in part by the European Commission via the projects ASSAS under Grant 101059682, in part by the ELIAS under Grant 101120237, in part by the INQUIRE under Grant 101057499, in part by the PARC under Grant 101057014, in part by the TAILOR under Grant 952215, and in part by the Slovenian Research Agency via the research program P2-0103 and the research Projects GC-0001, J1-3033, J2-2505, J2-4452, J2-4660, J3-3070, J4-3095, J5-4575, J7-4636, J7-4637, and N2-0236. *(Corresponding author: Marjan Stoimchev.)*

Marjan Stoimchev and Sašo Džeroski are with the Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia, and also with Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia (e-mail: marjan.stoimchev@ijs.si; saso.dzeroski@ijs.si).

Jurica Levatić and Dragi Kocev are with the Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia (e-mail: jurica.levatic@ijs.si; dragi.kocev@ijs.si).

Data are available online at: https://github.com/marjanstoimchev/SSL-MAE.
Digital Object Identifier 10.1109/JSTARS.2025.3578196

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

inherently *multilabel*, meaning that *multiple* classes typically coexist within a single image. Such oversimplification can lead to information loss and less accurate results, particularly when analyzing complex and heterogeneous landscapes.

Thus, to address both specific issues of RSI classification—the issue of labeled data scarcity and the presence of multiple semantic categories—semisupervised multilabel classification (MLC) methods are needed. Unfortunately, very few such methods for RSI classification were developed thus far [14]. Several semisupervised methods capable of solving the MLC task have been developed [15], [16], [17], with some addressing the complex visual contexts found in the computer vision domain [15], [16], [17]. However, none of these methods have been tailored to or rigorously evaluated in RS imagery, leaving a gap between the needs of the RSI domain and the available methods.

In an attempt to close this gap, in this article, we propose SSL-MAE: A novel method for multilabel semisupervised classification of RSIs. SSL-MAE is based on masked autoencoders (MAEs), a widely used self-supervised method, which relies on masked image modeling (MIM) as a pretext task for representation learning [18], [19]. They showed to be remarkably effective for solving many downstream tasks, including MLC [20], [21], [22] and are intrinsically suitable for addressing the limited labeled data problem [23]. Despite the apparent appropriateness of MAEs (and self-supervised methods in general) for SSL, they have one major drawback—the need for two-stage learning: pretraining on unlabeled data followed by fine-tuning on limited labeled data.

Our proposed approach, SSL-MAE, overcomes the limitations of two-stage learning approaches by *combining self-supervised learning and discriminative learning into a unified, end-to-end framework*. The information from the unlabeled data is utilized through self-supervised component based on a state-of-the-art model for representation learning (SimMIM) [19], while a discriminative classifier is simultaneously used to supervise the learning process by exploiting the limited labeled data. Both branches (self-supervised and discriminative) of the architecture share a common encoder.

The overall model is optimized with a learning objective that adaptively *combines* reconstruction loss (the unlabeled component) and classification loss (the labeled component). Within such a joint loss, we design a mechanism to control the amount of supervision and allow the method to adapt to the dataset at hand to either improve the predictive performance by exploiting unlabeled data or prevent performance degradation—the latter being a common pitfall of SSL. Lastly, the versatility of the design is not only tailored to one classification task, i.e., it can accommodate either MCC or MLC without significant changes to the learning objective.

The main contributions of our work are summarized as follows.

1) We propose a novel end-to-end semisupervised MLC learning method (i.e., SSL-MAE) that combines the strengths of discriminative learning and self-supervised learning while avoiding the need for two-step learning process.
2) We design a joint loss function that combines the supervised and the self-supervised loss terms into a single joint learning objective enabling SSL and adaptation of the amount of supervision to the dataset at hand.
3) We show that the proposed adaptive joint learning can be integrated with other state-of-the-art self-supervised methods, converting them into semisupervised end-to-end solutions and enhancing their predictive performance.
4) We perform rigorous evaluation and demonstrate that SSL-MAE successfully exploits unlabeled data and consistently outperforms the supervised baseline across all datasets and both learning tasks (MCC and MLC). We further demonstrate that SSL-MAE outperforms the competing state-of-the-art semisupervised methods while being among the top performers compared to the state-of-the-art self-supervised methods.

## II. Related Work

In this section, we discuss prior work in fields closely related to our study: self-supervised MAEs and semisupervised RSI classification.

### A. Masked Autoencoders

The state-of-the-art performance across a variety of domains obtained by using vision transformer (ViT) architecture has underscored MAEs as a predominant approach in the realm of self-supervised learning [18], [24]. They rely on MIM as a pretext task for learning robust representations [18], [19]. He et al. [18] showed that MAE are scalable self-supervised learners in the computer vision domain. Their architecture is designed to effectively train high-capacity models using unlabeled data while also achieving strong generalization across diverse downstream tasks through the integration of labeled data. This approach was further simplified with the introduction of SimMIM [19], which adopts a lightweight design for the decoder part of the network architecture that also shows strong generalization performance on downstream tasks. Both approaches have shown great potential in reducing the need for tedious manual labeling, which helps in achieving better performance than fully supervised methods. Moreover, leveraging high volumes of unlabeled data for representation learning with MAE, followed by fine-tuning with a smaller subset of labeled data, aligns well with the principles of SSL. Different from the above-mentioned two-stage approaches, Liang et al. proposed SupMAE [25], a more efficient approach to model training. Their approach extends SimMIM by adding a classification branch in parallel to the existing reconstruction branch. SupMAE can also be seen as a multiobjective learning approach where the model is trained with multiple auxiliary tasks. Their findings suggest that incorporating a supervised learning objective during the training phase enhances the model's robustness and its ability to learn better representations.

The versatility of MAE has also been recognized within the RS community, where considerable research effort has been directed toward RSI analysis [5], [26], [27], [28], [29]. Recently, Huang et al. have proposed CMAE [30], a novel approach that unifies contrastive learning (CL) and MIM through novel designs. Sun et al. introduced RingMO [29], a foundational model capable of addressing various downstream

tasks. Similar to [29], Jakubik et al. proposed Prithvi [31], which is a foundational MIM approach for modeling temporal data. They evaluated their approach on a wide range of tasks for Earth observation (EO) that have not been previously tackled by any foundational model, such as multitemporal cloud gap imputation, flood mapping, wildfire scar segmentation, and multitemporal crop segmentation.

### B. Semisupervised Learning

Within the field of SSL, certain approaches have become especially prominent. These include consistency regularization-based methods [8], [32], [33], pseudolabeling [9], [10], and self-training [34]. The main idea behind pseudolabeling and self-training is to create a new set of pseudolabels derived from the unlabeled data using confidence scores. Similarly, in cotraining approaches [35], [36], several models are trained concurrently on the same dataset. These models collaborate by sharing their predictions, and only the data points with unanimous agreement are employed in subsequent training iterations. Consistency regularization methods are designed to ensure stable output distributions, even when input images undergo random alterations. In the context of SSL, CL methods train models to identify whether samples are similar or dissimilar. This process involves reinforcing the consistency between various representations of the same sample, each subjected to different modifications. These approaches in SSL uniquely enhance the learning process by effectively utilizing both labeled and vast amounts of unlabeled data, and maximizing the advantages of each data type.

SSL gained considerable traction in the field of RSI classification. For instance, Kang et al. [13] proposed a deep metric learning architecture called HR-S $^2$ DML that utilizes a joint learning objective composed of normalized softmax loss with margin and a high-rankness regularization term. Pablo et al. [12] proposed MSMatch, which is an extended version of the well-known FixMatch methodology [8]. The authors showed that consistency regularization and pseudolabeling are very effective approaches to address the challenges of SSL in the RSI domain. While these methods deal with MCC scenarios, RSIs inherently require multilabel categorization, which allows for a more flexible representation of the inherent complexity present in the real world and avoids oversimplification of the classification process. However, acquiring multiple labels for RSIs is notably more challenging and time-consuming, further pronouncing the issue of limited labeled data availability. This underscores the critical demand for multilabel SSL methods tailored to effectively address these issues. To this end, Stoimchev et al. [37], proposed an effective approach utilizing predictive clustering trees and random forest ensembles thereof, to address the challenge of SSL for MLC.

According to the research discussed above, our proposed method, SSL-MAE, resembles the SimMIM approach [19] for the lightweight design of the decoder part of the model and SupMAE [25] for the use of the multiobjective learning. Unlike SimMIM's two-step process, i.e., self-supervised pretraining followed by fine-tuning, SSL-MAE addresses both tasks simultaneously through a combined objective that enhances representations and utilizes discriminative data from limited labeled samples. Moreover, different from SupMAE that only relies on the labeled samples in the joint objective with fixed weights, our approach also uses the unlabeled samples while dynamically balancing the weights of the two objectives to appropriately adapt the amount of supervision needed for the task at hand. This greatly improves predictive performance while also providing a safety mechanism to guard against the performance degradation than in some cases can be caused by the use of unlabeled data. Few other works incorporate joint learning with multiple objectives in the RS domain [38], [39] but none of them addresses the MLC problem.

## III. Methodology

A concise overview of the SSL-MAE pipeline is provided in Fig. 1, which comprises the following components:

1) *Image Masking* [see Fig. 1(a)]: Each input image, whether labeled or unlabeled, is partitioned into nonoverlapping patches. A fraction of these patches is randomly masked to create a reconstruction-based pretext task, driving the model to learn robust representations from partial visual data.
2) *Encoder* [see Fig. 1(b)]: A DeiT-based ViT architecture encodes the visible patches into latent embeddings. This step leverages the global modeling capabilities of self-attention, which is particularly effective for large-scale visual data.
3) *Classification Head* [see Fig. 1(c)]: Supervision is provided only by the latent representations corresponding to the unmasked patches in the labeled images. These partial embeddings are aggregated and passed to a lightweight classification head, ensuring the model's discriminative capability is driven solely by the available labeled data.
4) *Decoder* [see Fig. 1(d)]: A lightweight decoder reconstructs the pixel intensities of the masked patches, following a regression task for both labeled and unlabeled samples. This design aligns with SimMIM [19], providing a direct self-supervised signal to enhance feature learning.
5) *Joint learning* [see Fig. 1(e)]: Model parameters are updated via a weighted combination of supervised and unsupervised loss terms. This approach integrates the benefits of labeled supervision with the representation power gained from self-supervision on unlabeled data.

Subsequent sections detail these components, emphasizing their roles in enabling effective self-supervised learning on RSI.

### A. Image Masking

Consider a dataset $\mathcal{D} = \langle \mathcal{D}_L, \mathcal{D}_U \rangle$ that contains labeled images $\{(x_i^l, y_i^l)\}_{i=1}^{N_l}$ and unlabeled images $\{x_i^u\}_{i=1}^{N_u}$, where each $x_i^l, x_i^u \in \mathbb{R}^{C\times W\times H}$. Here, $C$ is the number of channels, and $W$ and $H$ are the width and height of the image, respectively. Each labeled or unlabeled image is first divided into $N_p = \frac{WH}{P^2}$ nonoverlapping patches, each of size $P \times P$ pixels, yielding $x_{p,i}^l, x_{p,i}^u \in \mathbb{R}^{N_p\times(C\times P^2)}$. Subsequently, a random masking operation is applied, where a fraction $m$ of the patches is withheld (masked). This masking step is crucial for creating a reconstruction-based self-supervised objective, inspired by the

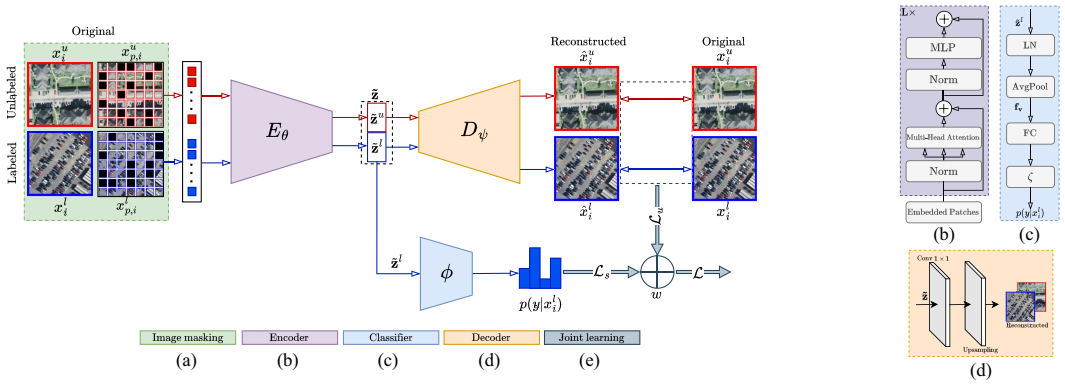



Fig. 1. Overview of SSL-MAE for SSL on RSIs. (a) labeled ($x^l$) and unlabeled ($x^u$) images are divided into patches, and a fraction of these patches is randomly masked. (b) ViT encoder processes both labeled and unlabeled patches to produce latent representations. The supervised objective $\mathcal{L}_s$ in (c) is computed solely on labeled (and unmasked) patches for classification. Meanwhile, the unsupervised objective $\mathcal{L}_u$ in (d) reconstructs the masked regions of both labeled and unlabeled images via an $\ell_1$ pixel-regression loss. In (e), the two losses are combined through a weight $w$, which regulates the degree of supervision during training. During inference, only the encoder (b) and classification head (c) are used on full (unmasked) test images.

MAE paradigm [18], which encourages the model to learn robust feature representations from partially visible data.

### B. Encoder

The encoder aims to map partially visible patches from both labeled and unlabeled images into a shared latent space. For simplicity, consider a batch containing one labeled and one unlabeled image. Each image is divided into $N_p$ patches (after masking), yielding $x_p^l \in \mathbb{R}^{N_p \times (C \times P^2)}$ and $x_p^u \in \mathbb{R}^{N_p \times (C \times P^2)}$. We then combine these along the patch dimension, forming $x \in \mathbb{R}^{2N_p \times (C \times P^2)}$. The process of mapping the input image $x$ to a latent feature representation $\tilde{\mathbf{z}}$ can be formally written as

$$\tilde{\mathbf{z}} = E\big(x, \theta\big): \quad \mathbb{R}^{2N_p \times (C \times P^2)} \longrightarrow \mathbb{R}^{2N_p \times d} \tag{1}$$

which can be separated into labeled $\tilde{\mathbf{z}}^l \in \mathbb{R}^{N_p \times d}$ and unlabeled $\tilde{\mathbf{z}}^u \in \mathbb{R}^{N_p \times d}$. Finally, the encoder's output is normalized with layer normalization [40], which stabilizes training and ensures consistent mean and variance [40], [41].

### C. Classification Head

Only the latent representations corresponding to the *visible* (unmasked) patches of labeled images, $\tilde{\mathbf{z}}^l$, are passed to a lightweight classification head. Concretely, we aggregate these labeled embeddings over the patch dimension using average pooling

$$\mathbf{f}_v = \frac{1}{N_p^l} \sum_{i=1}^{N_p^l} \tilde{\mathbf{z}}_v^{l(i)} \in \mathbb{R}^d \tag{2}$$

where $\tilde{\mathbf{z}}_v^{l(i)}$ denotes the embedding of the $i$th *visible* patch of labeled images, and $N_p^l$ is the total number of labeled patches retained after masking. The resulting feature vector, $\mathbf{f}_v$, is passed through a fully connected (FC) layer to generate the final class logits. A sigmoid activation function ($\zeta$) then processes these logits to produce the class probabilities, $p(y \mid x^l)$.

During training, for a labeled mini-batch of size $B_l$, we define the supervised loss $\mathcal{L}_s$ as

$$\mathcal{L}_s = \frac{1}{B_l} \sum_{i=1}^{B_l} \mathcal{H}\Big(y_i,\, p\big(y \mid x_i^l\big)\Big) \tag{3}$$

where $y_i$ is the ground-truth label for the $i$th labeled sample, and $\mathcal{H}$ denotes the task-dependent loss function [e.g., cross-entropy (CE) for MCC or binary CE for MLC].

### D. Decoder

Following SimMIM [19], we employ a *lightweight decoder* $D(\cdot, \psi)$ to reconstruct the raw pixel values of the masked patches from the encoder embeddings

$$\hat{x} = D\big(\tilde{\mathbf{z}}, \psi\big) \in \mathbb{R}^{C \times W \times H} \tag{4}$$

where $\psi$ denotes the decoder's learnable parameters. A $1 \times 1$ convolution first maps the latent features into a lower resolution prediction, which is then upsampled via an efficient subpixel convolution layer to match the original spatial dimensions.

We train the decoder on both labeled and unlabeled images via an $\ell_1$ pixel-reconstruction loss computed only over the masked regions

$$\mathcal{L}_u = \frac{1}{B} \sum_{i=1}^{B} \left( \frac{1}{\Omega(x_{M_i})} \sum_{m \in M_i} \big\| \hat{x}_{i,m} - x_{i,m} \big\|_1 \right) \tag{5}$$

where $B = B_l + B_u$ is the total batch size (labeled plus unlabeled), $M_i$ denotes the set of masked pixels for the $i$th image, $\Omega(\cdot)$ is the number of elements, and $\|\cdot\|_1$ is the $L_1$ norm operator.

Although the reconstruction objective is critical for guiding the model to learn robust feature representations, our *primary focus* remains on the classification task. Consequently, we also add a *discriminative branch* that leverages the visible patches from labeled images (see Section III-C). In other words, the reconstruction loss predominantly refines the learned feature

representations, while the classification head remains the primary factor influencing overall performance.

### E. Adaptive Joint Learning

To jointly optimize for both supervised (discriminative) and self-supervised (reconstruction-based) objectives, we introduce a weighted sum of the corresponding losses

$$\mathcal{L} = w\,\mathcal{L}_s \;+\; (1-w)\,\mathcal{L}_u \tag{6}$$

where $w \in [0,1]$ balances the emphasis on the supervised loss $\mathcal{L}_s$ versus the unsupervised loss $\mathcal{L}_u$. We investigate two strategies for choosing $w$.

1) *Grid Search (GS) Optimization:* We train separate models for various fixed values of $w$ from a candidate set $W = \{w_1, w_2, \ldots, w_n\}$. The model achieving the best performance metric on a validation set determines $w^* = \arg\max_{w_i \in W} P(w_i)$, which is then evaluated on the test set.
2) *Learnable weight:* We parameterize $w$ as a trainable parameter $w_\ell = \sigma(\theta_\ell)$, where $\sigma(\theta_\ell) = \frac{1}{1+e^{-\theta_\ell}}$ is the sigmoid function restricting $w_\ell$ to [0,1]. Substituting into (6), the final loss function becomes:

$$\mathcal{L}(\theta, \theta_\ell) = \sigma(\theta_\ell)\,\mathcal{L}_s(\theta) \;+\; (1-\sigma(\theta_\ell))\,\mathcal{L}_u(\theta) \tag{7}$$

where $\theta$ represents all network parameters. During training, both $\theta$ and $\theta_\ell$ are updated via gradient descent

$$\theta \leftarrow \theta - \eta\,\frac{\partial \mathcal{L}}{\partial \theta}, \quad \theta_\ell \leftarrow \theta_\ell - \eta\,\frac{\partial \mathcal{L}}{\partial \theta_\ell} \tag{8}$$

with $\frac{\partial \mathcal{L}}{\partial \theta_\ell} = \sigma(\theta_\ell)(1-\sigma(\theta_\ell))[\mathcal{L}_s(\theta) - \mathcal{L}_u(\theta)]$.

When $w_\ell \to 1$, the model converges to a fully supervised solution; when $w_\ell \to 0$, it operates in a purely self-supervised regime. By learning $w_\ell$ jointly with $\theta$, the network automatically adjusts its reliance on labeled versus unlabeled data based on the relative magnitudes of $\mathcal{L}_s$ and $\mathcal{L}_u$. This adaptive scheme allows SSL-MAE to effectively leverage abundant unlabeled data in early training stages while gradually emphasizing discriminative learning as labeled samples become more informative, thereby improving performance on RS tasks with limited annotations.

## IV. Experimental Design

### A. Datasets

We assess the performance of the proposed methods on 10 publicly available land use and land cover RSI datasets: 5 datasets for MCC and 5 datasets for MLC. We selected these datasets based on their diversity along several dimensions, including image resolution, number of labeled images available, and geographical location.

*1) MCC Datasets:* We used the following 5 datasets for the MCC task: OPTIMAL-31 [42], UCM [43], AID [44], RSSCN7 [45], and RESISC45 [46]. OPTIMAL-31 contains 31 distinct categories for scene classification, with images collected from Google Earth and with a pixel resolution of $256 \times 256$. UCM Land Use contains 2100 images grouped into 21 broad categories at the scene level with a resolution of $256 \times 256$ pixels. AID on the other hand is a more challenging dataset than UCM containing 3000 aerial images grouped into 30 object categories and higher resolution images of $600 \times 600$ pixels. The RSSCN7 dataset consists of 2800 scene images obtained from Google Earth, with a resolution of $400 \times 400$. Lastly, the RESISC45 dataset created by Northwestern Polytechnical University (NWPU) has 31 500 images with $256 \times 256$ resolution and covers 45 scene classes.

*2) MLC Datasets:* For the MLC task, we used a total of five datasets: the MLC versions of UCM [47] and AID [48] datasets and three more datasets originally developed for the MLC task, namely, DFC-15 [49], MLRSNet [50] and the BigEarthNet dataset (BEN-43) [51]. The MLC version of UCM and AID datasets contains 17 object-level labels. On the other hand, DFC-15 is built from the 2015 IEEE GRSS data fusion contest, created from a semantic segmentation dataset. It contains a total of 3342 image patches with a resolution of $600 \times 600$ pixels, where the image patches are grouped into 8 different object categories. The MLRSNet dataset contains 109 161 images annotated into 60 categories, with an image resolution of $256 \times 256$ pixels. Lastly, we use the BEN-43, which is one of the largest datasets to date, containing 590 326 Sentinel-2/S2 (and later also Sentinel-1/S1) image patches acquired over Europe (Austria, Belgium, Finland, Ireland, Kosovo, Lithuania, Luxembourg, Portugal, Serbia, Switzerland) and developed by remote sensing image analysis (RSiM) Group and the Database Systems and Information Management Group at the Technische Universität Berlin (TU Berlin). For the experiments, we use three of the S2 channels (RGB images) and the original CORINE Land Cover nomenclature with 43 labels. To ensure that the dataset is comparable in size to the other datasets used in our study and to avoid issues with computational times, we sampled (using stratification) approximately 1% of the BEN-43. The final dataset contains 5903 images.

### B. Implementation Details

For the backbone of SSL-MAE, we use the base version of DeiT [52], specifically DeiT-B/16, which utilizes a patch resolution of $16 \times 16$ pixels. The model is initialized with weights pretrained on ImageNet [53]. All models are trained on RGB images with a resolution of $224 \times 224$ using the combined learning objective described in (6). The default training configuration is summarized in Table I. The dimension of the latent representation, obtained after applying average pooling with a kernel size of 1, is $d = 768$.

Unlike the original SimMIM implementation [19], our approach uses a lower masking ratio of 0.3. This design choice favors classification performance over image reconstruction, with the reconstruction loss primarily serving to guide the model in generating improved representations for the classification task.

To manage the imbalance between labeled and unlabeled images in the dataset, where the unlabeled portion is significantly larger, we perform random sampling of the labeled subset in every training epoch. This sampling ensures that the number of labeled and unlabeled samples remains equal during training

TABLE I
CONFIGURATION PARAMETERS OF SSL-MAE AND THEIR VALUES

| Configuration | Value |
|---|---|
| Optimizer | AdamW |
| Base learning rate | 1e 3 |
| Minimum learning rate | 1e 5 |
| Weight decay | 1e 6 |
| Optimizer momentum | $\beta_1 = 0.9$, $\beta_2 = 0.999$ |
| Batch size | 64 |
| Learning rate schedule | Cosine decay |
| Warmup epochs | 5 |
| Gradient accumulation steps | 5 |
| Total training epochs | 100 |
| Type of data augmentations | Strong |
| Masking ratio | 0.3 |

iterations. As a result, both types of data contribute equally to the optimization process.

We apply strong data augmentations to all images, regardless of their label availability. This approach makes it highly unlikely for the model to see the exact same image more than once, even across multiple epochs. Such a strategy not only promotes generalization but also serves as a form of implicit regularization, thereby helping to prevent overfitting on the limited labeled data. The sampled data is revisited in every epoch to ensure balanced training.

The same masking ratio is consistently applied across all evaluated datasets. In addition to masking, which regularizes the model by exposing only a random subset of visible patches to the classification branch, our augmentation strategy is more aggressive compared to previous methods [25], [54], which often employ milder augmentations for labeled data.

To study the influence of supervised learning, we tune the weighting parameter $w$ over a wide range depending on the dataset. In the SSL-MAE-GS version, the value of $w$ is varied between 0 and 1in increments of 0.1. For the SSL-MAE-$w_\ell$ variant, the weighting coefficient $w_\ell$ is automatically learned during training.

All experiments are conducted using a single NVIDIA A100 GPU with 80 GB of memory. Each training run spans 100 epochs and employs the AdamW optimizer with a base learning rate of $1 \times 10^{-3}$. We use cosine learning rate scheduling, a mini-batch size of 64, and mixed precision training to enhance computational efficiency and reduce memory usage.

### C. Evaluation Strategy

To assess the efficacy of the SSL methods, we adopt the inductive learning setting, aiming to train models capable of predicting labels for new training-unseen examples. This involves using a distinct, fixed-size test set for evaluating the performance of the final model. We used predetermined test sets where available, while in the other cases, we divided the datasets into training and testing parts through an iterative stratified sampling strategy to preserve the relative frequency of the labels in the datasets. The resulting train and test ratios are approximately 80% and 20%, and the validation set is approximately 10% partitioned from the training set.

We explore the relationship between model performance and the proportion of labeled data by using varying percentages of labeled data from the training sets. Labeled samples were randomly selected from the training set in proportions of $1, 5, 10, 25$ percent. The remaining training samples were treated as unlabeled by removing their labels. Each percentage of unlabeled data was subjected to three separate experiments with a different random initialization. The overall predictive performance is the average result from these three iterations.

We compare our proposed methods with supervised and semisupervised baseline methods formulated as follows. The supervised baseline can be easily derived from the SSL-MAE by only setting the $w$ parameter to 1, which means optimizing the model parameters solely by using the supervised learning objective $\mathcal{L}_s$ (3). The semisupervised baseline is a simplistic formulation based on a self-supervised pretraining step on the unlabeled data, followed by a supervised fine-tuning step on the remaining labeled data. To achieve this, our method needs to be used in a two-stage manner while applying several crucial modifications. In the first stage, the $w$ parameter is set to 0, and the model is learned on the unlabeled data. This means that the model parameters are optimized through the unsupervised learning objective $\mathcal{L}_u$ (5) through self-supervision. Once the pretraining stage is done, the model is applied to the labeled data in the same way as in the supervised baseline setting described above, but this time the model has some stored knowledge about the underlying structure of the unlabeled data.

We also compare the proposed methods with several state-of-the-art semisupervised and self-supervised methods. We used semisupervised methods from the Unified SSL Benchmark[1] [55]: FixMatch [8], CoMatch [56], FlexMatch [9], SoftMatch [57], FreeMatch [33], and SimMatch [58]. The self-supervised approaches we selected from the Lightly framework[2] [59]: SimCLR [60], BYOL [61], DINO [24], VICReg [62], and MOCO [63]. Moreover, we also included a recent state-of-the-art self-supervised method from the RS domain called Tri-ReD [64]. This method introduces a novel trilateral loss and tri-branch augmentation strategy, demonstrating strong performance on urban scene classification tasks. We set each method's hyperparameters according to the specifications their original authors recommend.

### D. Evaluation Measures

We employ two distinct performance metrics to evaluate our method across different classification scenarios: the area under precision-recall curve (AUPRC) for MCC and the Area Under microaveraged precision-recall curve (AU$\overline{\text{PRC}}$) for MLC. Both metrics are threshold-independent, providing reliable assessment across various operating points.

*AUPRC for MCC:* In the multiclass setting, where each image is assigned to exactly one class, we compute the standard AUPRC. For each class $i$, precision and recall are calculated at various threshold values

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{9}$$

[1][Online]. Available: https://github.com/microsoft/Semi-supervised-learning

[2][Online]. Available: https://github.com/lightly-ai/lightly

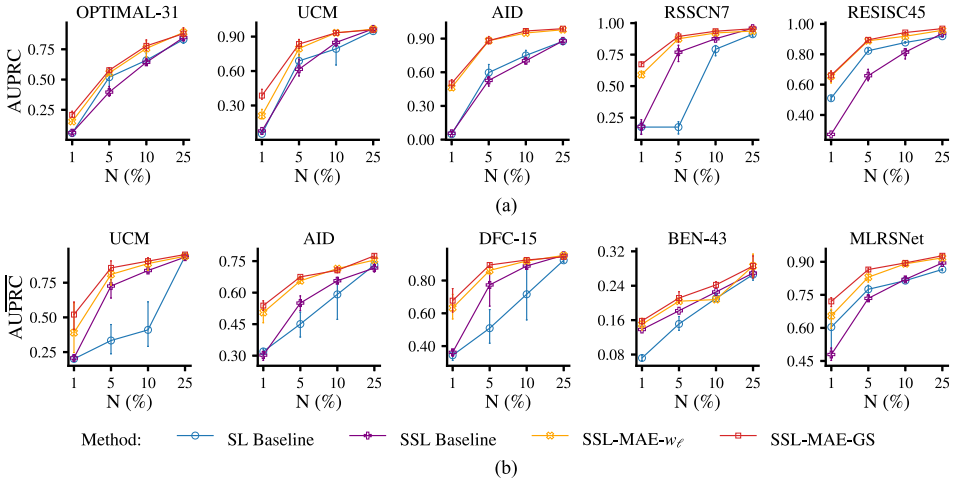



Fig. 2. Predictive performance of supervised and semisupervised methods on MCC and MLC datasets of RSI. The learning curves depict $\mu \pm \sigma$ (mean and standard deviation) for AUPRC and $\overline{\text{AUPRC}}$ across three repeats, where higher values indicate better predictive performance. Note that the $\sigma$ values are typically very small and are not visible in the graphs (especially at higher percentages of labeled data. (a) MCC datasets. (b) MLC datasets.

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{10}$$

where $\text{TP}_i$, $\text{FP}_i$, and $\text{FN}_i$ represent true positives, false positives, and false negatives for class $i$, respectively. The AUPRC is then computed as the area under the precision–recall curve across varying threshold values. In multiclass scenarios, we report the macroaveraged AUPRC, which gives equal weight to each class regardless of its frequency in the dataset.

*AU$\overline{\text{PRC}}$ for MLC:* For MLC, where each image can belong to multiple classes simultaneously, we use AU$\overline{\text{PRC}}$, which employs microaveraging across all classes. The microaveraged precision and recall are calculated as

$$\overline{\text{Prec}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FP}_i} \tag{11}$$

$$\overline{\text{Rec}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FN}_i} \tag{12}$$

where the summation over $i$ iterates through all classes. Unlike macroaveraging, microaveraging aggregates the contributions of all classes before calculating the metric, which accounts for class imbalance and better represents overall performance when classes can overlap. The AU$\overline{\text{PRC}}$ is then computed as the area under the precision–recall curve when evaluating microaveraged precision against microaveraged recall across varying threshold values.

The key difference between these metrics lies in how they handle the fundamentally different nature of classification tasks: AUPRC treats each class independently in a mutually exclusive setting, while AU$\overline{\text{PRC}}$ accounts for potential overlap between classes by using microaveraging across all classes simultaneously.

To compute both metrics, we use the implementations available in `scikit-learn` [65].

## V. Results

### A. Comparison With Supervised Learning

For SSL to be meaningful, it has to perform better than a corresponding supervised algorithm that uses only labeled data [66]. To provide such a proof-of-principle for our proposed SSL methods, we compare their performance with the performance of supervised and semisupervised baseline methods [for their definition (see Section IV-C)].

The results presented in Fig. 2 demonstrate that the proposed methods, SSL-MAE-GS and SSL-MAE-$w_l$, successfully exploit unlabeled data: They consistently outperform, often by a large margin, both supervised and semisupervised baselines across all datasets, different fractions of labeled data, and both MCC and MLC learning tasks.

As expected, the advantage of the proposed SSL methods is the largest for smaller percentages of labeled data ($<10\%$ ) and it slowly decreases as the amount of labeled data increases. The semisupervised baseline achieves some advantage over the supervised baseline for some of the datasets, however, contrary to the proposed methods, it also sometimes degrades the performance in comparison to the supervised baseline (e.g., for MLRSNet, AID MCC, and RESIC45 datasets).

Out of the two proposed variants of our method, SSL-MAE-GS consistently achieves slightly better results than SSL-MAE-$w_l$. This performance advantage, however, comes with the cost of increased computational time: To optimize the $w$ parameter, SSL-MAE-GS trains a model for each candidate $w$ value. In Fig. 3, we show the learning times needed for the largest MCC and MLC datasets. The results show that SSL-MAE-$w_l$ is approximately one order of magnitude faster than SSL-MAE-GS, offering a good tradeoff between performance and speed. Naturally, both algorithms are slower than the supervised baseline, since they deal with a much larger number of examples (unlabeled in addition to labeled ones).

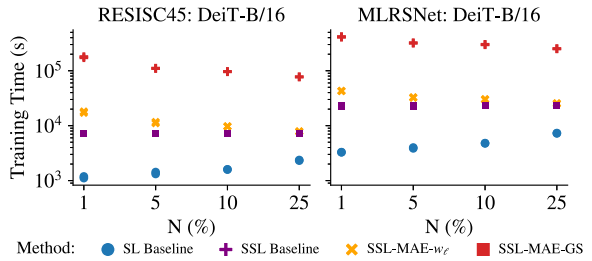



Fig. 3. Training times of different methods evaluated on RESISC45 for MCC and MLRSNet for MLC.

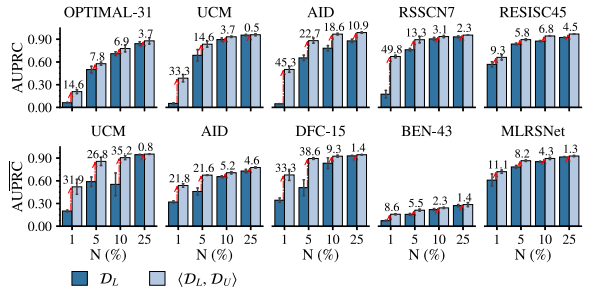



Fig. 4. Influence of the unlabeled data on the predictive performance of SSL-MAE-GS. The results are in terms of AUPRC for MCC and $\overline{\text{AUPRC}}$ for MLC, where the performance improvements when using both labeled and unlabeled data ($\langle \mathcal{D}_L, \mathcal{D}_U \rangle$) over the labeled data only ($\mathcal{D}_L$), are highlighted with red arrows and the numbers depict the percentages of improvement.

### B. Influence of the Unlabeled Data

Having successfully established that the proposed SSL methods outperform their supervised counterparts, we next investigate whether the advantage in the predictive performance is due to the successful exploitation of unlabeled data, or due to the adaptive joint loss learning we introduced (see Section III-E). We thus compare the proposed SSL-MAE-GS method with its supervised variant that uses adaptive joint loss learning but does not consider unlabeled examples. This enables us to precisely measure the influence of unlabeled data on the predictive performance since the two methods are algorithmically identical, with the only difference being in the exploitation of the unlabeled data.

The results of this investigation are presented in Fig. 4—they clearly show that our method successfully exploits unlabeled data. Namely, SSL-MAE-GS consistently outperforms its supervised variant across all datasets, labeled fractions, and MCC and MLC learning tasks. Similarly, as we noted before, this advantage decreases as the amount of labeled data increases.

### C. Controlling the Amount of Supervision Through the $w$ Parameter

We next investigate the importance of controlling and adapting the amount of supervision in the process of learning using the joint loss. Recall that this is achieved via the $w$ parameter that controls the amount of supervision used to learn the models, ranging from unsupervised ($w = 0$), semisupervised ($0 < w < 1$), to fully supervised ($w = 1$). We, thus, investigate results for $w$ ranging from 0 to 1, given in Table II. We can observe that the optimal $w$ varies among different datasets and fractions of labeled data, emphasizing the importance of the adaptability of the proposed methods. Note that, the SSL-MAE-GS correctly selects the optimal $w$ value in most of the cases, especially for larger MCC and MLC datasets.

As it is well known from the literature, SSL does not always improve over supervised learning, thus it is important to exploit unlabeled data safely (see[67]). Within our method, this is enabled by the adaptability of joint loss learning via the $w$ parameter. For example, for the RESISC45 dataset and $>1\%$ labeled of labeled data, our algorithm correctly reverted to supervised learning (by selecting $w = 1$) and prevented performance degradation unlabeled data would otherwise cause.

### D. Comparison to the State-of-the-Art Methods

We next compare the performance of the proposed methods with the performance obtained by using state-of-the-art methods from the literature (listed in Section IV-C). We perform the comparison using several semisupervised methods and additionally with several methods from the related self-supervised learning field. Note that we performed the comparison with state-of-the-art semisupervised methods only on the MCC datasets since their implementation is tailored for the MCC task and they cannot be directly applied to MLC datasets. In contrast, the self-supervised methods have a more versatile design, enabling their application across MCC and MLC tasks.

The experimental analysis reveals that our methods are superior to the existing semisupervised state-of-the-art methods (see Table III). Our methods achieve better predictive performance (except in a few cases) across all datasets and fractions of labeled data considered. According to the average ranks given in the last part of Table III, SSL-MAE-GS is the best overall performer outranking the state-of-the-art methods by a large margin, closely followed by SSL-MAE-$w_l$. We also note that for $>1\%$ of labeled data, most of the semisupervised state-of-the-art methods perform worse than the supervised baseline, further highlighting the importance of the supervision adaptability introduced in our methods, which can prevent the performance degradation if the method fails to exploit unlabeled data.

The trend of the superiority of our methods is, however, not fully preserved in comparison to the state-of-the-art self-supervised methods (see Table IV). Our proposed method, SSL-MAE-GS, is the best performer in scenarios with few labeled examples available: 1% and 5% for MCC datasets, and 1% for MLC datasets. As the amount of labeled data increases above 5%, BYOL and MoCo methods emerge as top two performers for both MCC and MLC datasets. The Tri-ReD method also exhibits good performance, albeit only for MLC datasets, whereas for MLC datasets it underperformed.

### E. General Utility of Adaptive Joint Loss Learning

Motivated by the results obtained in the previous section, here we investigate whether the proposed adaptive joint learning process can be used to enhance methods outside of the MAE framework used in the proposed algorithm. Namely, in comparison with self-supervised methods, our proposed method

TABLE II
INFLUENCE OF THE AMOUNT OF SUPERVISION OF SSL-MAE-GS

| MCC | OPTIMAL-31 N (%) | | | | UCM N (%) | | | | AID N (%) | | | | RSSCN7 N (%) | | | | RESISC45 N (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| 0.0 | 0.057 | 0.070 | 0.055 | 0.064 | 0.060 | 0.068 | 0.058 | 0.077 | 0.056 | 0.063 | 0.060 | 0.063 | 0.146 | 0.148 | 0.150 | 0.152 | 0.034 | 0.032 | 0.032 | 0.032 |
| 0.1 | 0.183 | 0.577 | 0.760 | 0.869 | 0.334 | 0.797 | 0.929 | 0.930 | 0.305 | 0.751 | 0.867 | 0.862 | 0.623 | 0.845 | 0.884 | 0.930 | 0.644 | 0.889 | 0.927 | 0.945 |
| 0.2 | 0.176 | **0.585** | **0.767** | 0.880 | 0.346 | **0.826** | 0.918 | 0.961 | 0.280 | 0.740 | 0.860 | 0.884 | 0.654 | 0.875 | 0.928 | 0.934 | 0.657 | 0.886 | 0.923 | 0.941 |
| 0.3 | 0.201 | 0.542 | 0.755 | 0.874 | 0.329 | 0.789 | 0.904 | 0.941 | 0.282 | 0.713 | 0.859 | 0.910 | 0.620 | 0.847 | 0.913 | 0.940 | 0.624 | 0.879 | 0.929 | 0.942 |
| 0.4 | 0.126 | 0.552 | 0.752 | 0.855 | 0.324 | 0.809 | 0.897 | 0.956 | 0.285 | 0.730 | 0.838 | 0.909 | **0.658** | **0.900** | 0.910 | 0.954 | 0.604 | 0.879 | 0.929 | 0.940 |
| 0.5 | 0.154 | 0.553 | 0.758 | 0.873 | 0.212 | 0.796 | **0.932** | 0.961 | **0.501** | **0.882** | **0.968** | 0.986 | 0.556 | 0.883 | **0.936** | 0.951 | **0.660** | 0.890 | 0.926 | 0.958 |
| 0.6 | 0.149 | 0.535 | 0.744 | 0.880 | 0.254 | 0.809 | 0.892 | 0.950 | 0.284 | 0.734 | 0.807 | 0.916 | 0.585 | 0.861 | 0.935 | 0.951 | 0.624 | 0.879 | 0.929 | 0.943 |
| 0.7 | **0.182** | 0.535 | 0.741 | 0.878 | 0.354 | 0.788 | 0.892 | 0.948 | 0.262 | 0.735 | 0.808 | 0.916 | 0.552 | 0.882 | 0.935 | 0.951 | 0.618 | 0.884 | 0.927 | 0.940 |
| 0.8 | 0.170 | 0.535 | 0.740 | 0.877 | **0.359** | 0.788 | 0.892 | 0.951 | 0.264 | 0.733 | 0.807 | 0.915 | 0.544 | 0.858 | 0.933 | **0.952** | 0.629 | 0.875 | 0.928 | 0.938 |
| 0.9 | 0.161 | 0.533 | 0.740 | 0.876 | 0.325 | 0.788 | 0.890 | 0.951 | 0.307 | 0.734 | 0.807 | 0.916 | 0.604 | 0.870 | 0.934 | 0.952 | 0.630 | 0.879 | 0.929 | 0.936 |
| 1.0 | 0.062 | 0.062 | 0.062 | **0.889** | 0.053 | 0.688 | 0.793 | **0.966** | 0.048 | 0.814 | 0.903 | **0.988** | 0.175 | 0.743 | 0.881 | 0.907 | 0.027 | **0.892** | **0.944** | **0.969** |

| MLC | UCM N (%) | | | | AID N (%) | | | | DFC-15 N (%) | | | | BEN-43 N (%) | | | | MLRSNet N (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $w$ | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| 0.0 | 0.225 | 0.215 | 0.207 | 0.232 | 0.320 | 0.331 | 0.330 | 0.322 | 0.357 | 0.361 | 0.355 | 0.367 | 0.078 | 0.080 | 0.078 | 0.077 | 0.100 | 0.100 | 0.099 | 0.099 |
| 0.1 | 0.473 | 0.773 | 0.834 | 0.909 | 0.472 | 0.575 | 0.654 | 0.689 | 0.677 | 0.832 | 0.878 | 0.926 | 0.148 | 0.194 | 0.224 | 0.270 | 0.572 | 0.811 | 0.867 | 0.923 |
| 0.2 | 0.493 | 0.833 | 0.899 | 0.938 | 0.515 | 0.643 | 0.703 | 0.716 | 0.640 | 0.815 | 0.880 | 0.922 | 0.151 | 0.201 | 0.227 | 0.273 | 0.594 | 0.797 | 0.856 | 0.913 |
| 0.3 | **0.494** | **0.852** | **0.907** | 0.940 | 0.525 | 0.662 | 0.686 | 0.749 | 0.665 | 0.861 | 0.888 | 0.932 | 0.146 | 0.193 | 0.231 | 0.286 | 0.691 | 0.817 | 0.842 | 0.913 |
| 0.4 | 0.513 | 0.841 | 0.887 | 0.933 | **0.528** | **0.658** | 0.693 | 0.760 | 0.657 | **0.896** | 0.907 | 0.915 | 0.150 | 0.204 | 0.226 | 0.290 | 0.683 | 0.824 | 0.863 | 0.909 |
| 0.5 | 0.392 | 0.830 | 0.902 | 0.946 | 0.508 | 0.651 | 0.700 | **0.764** | 0.670 | 0.854 | 0.905 | 0.928 | 0.149 | 0.207 | 0.232 | 0.277 | 0.593 | **0.865** | **0.890** | 0.921 |
| 0.6 | 0.491 | 0.789 | 0.896 | 0.947 | 0.455 | 0.611 | **0.750** | 0.760 | 0.598 | 0.887 | 0.899 | 0.939 | 0.146 | 0.209 | 0.232 | **0.301** | 0.702 | 0.831 | 0.868 | 0.891 |
| 0.7 | 0.491 | 0.789 | 0.896 | 0.951 | 0.526 | 0.613 | 0.693 | 0.769 | **0.681** | 0.892 | **0.917** | 0.918 | 0.150 | **0.211** | 0.240 | 0.285 | 0.709 | 0.850 | 0.880 | 0.888 |
| 0.8 | 0.492 | 0.789 | 0.898 | 0.952 | 0.538 | 0.592 | 0.716 | 0.768 | 0.659 | 0.889 | 0.914 | **0.940** | 0.156 | 0.185 | 0.238 | 0.278 | **0.709** | 0.847 | 0.878 | 0.882 |
| 0.9 | 0.492 | 0.788 | 0.899 | **0.954** | 0.540 | 0.592 | 0.715 | 0.777 | 0.634 | 0.892 | 0.916 | 0.935 | **0.158** | 0.183 | **0.242** | 0.279 | 0.707 | 0.853 | 0.885 | 0.878 |
| 1.0 | 0.201 | 0.333 | 0.410 | 0.944 | 0.320 | 0.450 | 0.592 | 0.747 | 0.344 | 0.760 | 0.873 | 0.919 | 0.072 | 0.151 | 0.211 | 0.266 | 0.587 | 0.833 | 0.888 | **0.923** |

The AUPRC measure is used for the MCC datasets and the AU$\overline{\text{PRC}}$ measure for MLC datasets where higher values indicate better predictive performance. For each dataset and percentage of labeled data, the best performing $w$ is highlighted in bold, while the $w$ value selected by the SSL-MAE-GS is coloured: green denotes that SSL-MAE-GS selected the optimal value of $w$, while red denotes suboptimal selection. To save space, the results are presented from a single out of the three random runs, since findings are similar across all runs.

TABLE III
PREDICTIVE PERFORMANCE OF THE PROPOSED SEMISUPERVISED METHODS AS COMPARED AGAINST STATE-OF-THE-ART SEMISUPERVISED METHODS FROM THE LITERATURE

| MCC Datasets | OPTIMAL-31 N (%) | | | | UCM N (%) | | | | AID N (%) | | | | RSSCN7 N (%) | | | | RESISC45 N (%) | | | | Avg. ranks N (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| SL Baseline | 0.062 | 0.450 | 0.610 | **0.889** | 0.053 | 0.688 | 0.793 | **0.966** | 0.048 | 0.814 | 0.903 | **0.988** | 0.175 | 0.743 | 0.881 | 0.907 | 0.027 | 0.892 | **0.944** | **0.969** | 9.0 | 5.0 | 5.1 | 3.0 |
| SoftMatch | 0.092 | 0.274 | 0.419 | 0.512 | 0.147 | 0.549 | 0.806 | 0.944 | 0.091 | 0.309 | 0.625 | 0.720 | 0.340 | 0.699 | 0.848 | 0.920 | 0.217 | 0.514 | 0.632 | 0.868 | 7.2 | 8.1 | 7.7 | 7.6 |
| FixMatch | 0.097 | 0.331 | 0.628 | 0.846 | 0.142 | 0.620 | 0.851 | 0.955 | 0.081 | 0.347 | 0.678 | 0.916 | 0.356 | 0.717 | 0.874 | 0.942 | 0.300 | 0.821 | 0.895 | 0.948 | 6.6 | 6.4 | 5.6 | 4.7 |
| FreeMatch | 0.099 | 0.251 | 0.372 | 0.529 | 0.166 | 0.535 | 0.755 | 0.946 | 0.105 | 0.307 | 0.630 | 0.844 | 0.346 | 0.748 | 0.833 | 0.909 | 0.212 | 0.522 | 0.634 | 0.842 | 6.2 | 7.4 | 8.2 | 7.4 |
| SimMatch | 0.093 | 0.417 | 0.630 | 0.636 | 0.207 | 0.664 | 0.871 | 0.954 | 0.125 | 0.503 | 0.787 | 0.921 | 0.359 | 0.746 | 0.879 | 0.932 | 0.492 | 0.767 | 0.907 | 0.551 | 5.0 | 5.2 | 4.6 | 6.3 |
| FlexMatch | 0.102 | 0.339 | 0.419 | 0.498 | 0.228 | 0.572 | 0.812 | 0.955 | 0.122 | 0.392 | 0.652 | 0.816 | 0.395 | 0.747 | 0.859 | 0.932 | 0.192 | 0.514 | 0.629 | 0.853 | 4.8 | 6.3 | 7.1 | 6.8 |
| CoMatch | **0.220** | 0.505 | 0.739 | 0.868 | 0.285 | 0.696 | 0.904 | 0.941 | 0.193 | 0.639 | 0.826 | 0.934 | 0.521 | 0.788 | 0.888 | 0.952 | 0.587 | 0.885 | 0.918 | 0.954 | 2.4 | 3.4 | 3.4 | 4.6 |
| SSL-MAE-$w_\ell$ | 0.153 | 0.555 | 0.757 | **0.889** | 0.212 | 0.797 | **0.933** | **0.966** | 0.463 | 0.881 | 0.950 | 0.979 | 0.590 | 0.876 | 0.922 | 0.937 | 0.656 | 0.889 | 0.919 | 0.958 | 2.6 | 2.2 | 2.1 | 2.6 |
| SSL-MAE-GS | 0.208 | **0.577** | **0.778** | 0.879 | **0.386** | **0.835** | **0.933** | 0.960 | **0.501** | **0.882** | **0.968** | **0.988** | **0.673** | **0.894** | **0.936** | **0.955** | **0.661** | **0.895** | **0.944** | **0.969** | **1.2** | **1.0** | **1.2** | **2.0** |

The results are presented using the AUPRC measure for MCC datasets (higher values indicate better predictive performance). The last set of columns denotes average ranks across datasets. For each dataset and percentage of labeled data, the best performance is highlighted in bold, while the second best is underlined.

SSL-MAE-GS was the best performer for small amounts of labeled data, while BYOL and MoCo methods performed better for larger amounts of labeled data. We opted to explore if introducing the adaptive joint learning to BYOL and MoCo would mitigate their weaknesses and yield methods that work well across both small and large amounts of labeled data.

To this end, we integrated adaptive joint learning into BYOL and MoCo. Due to its efficiency in terms of computational time, we used the $w_l$ version of the algorithm which treats the $w$ parameter as a learnable weight. The results, presented in Fig. 5 show that incorporating adaptive joint learning greatly enhances the predictive performance of BYOL and MoCo for smaller percentages of labeled data ($<$10% ) on both MCC and MLC datasets. The only exception is the RESISC45 dataset, where the modification did not yield improvements. This is by far the largest MCC dataset we considered, thus, possibly even 1% and 5% fractions already contain enough labeled examples for methods to achieve high predictive performance. Similarly, the improvements are the least pronounced for the largest MLC dataset—MLRSNet.

To highlight the magnitude of improvement, we further present the results in terms of relative differences illustrated with the heatmaps given in Fig. 6. On average, for 1% and 5% of labeled data, the addition of the adaptive joint learning strategy brings 300% –400% relative improvement in the predictive performance for BYOL and 400% –500% for MoCo for MCC datasets, and 55% –95% and 45% –70% improvement for MLC datasets, respectively. In contrast, for larger amounts of labeled data where joint learning strategy does not exhibit improvements, the relative performance differences are negligible, showing that our proposed adaptive joint learning can successfully enhance other methods by greatly improving their weaknesses while preserving their strengths.

### F. Qualitative Evaluation

This section presents the qualitative evaluation of the proposed SSL-MAE method. For simplicity of the analysis, we only used the MCC and MLC versions of the AID dataset.

TABLE IV
PREDICTIVE PERFORMANCE OF THE PROPOSED SEMISUPERVISED METHODS AS COMPARED AGAINST STATE-OF-THE-ART SELF-SUPERVISED METHODS USED FROM THE LITERATURE

| MCC Datasets | OPTIMAL-31 N (%) | | | | UCM N (%) | | | | AID N (%) | | | | RSSCN7 N (%) | | | | RESISC45 N (%) | | | | Avg. ranks N (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| SL Baseline | 0.062 | 0.450 | 0.610 | 0.889 | 0.053 | 0.688 | 0.793 | 0.966 | 0.048 | 0.814 | 0.903 | **0.988** | 0.175 | 0.743 | 0.881 | 0.907 | 0.027 | 0.892 | 0.944 | 0.969 | 10.0 | 5.4 | 6.0 | 3.8 |
| SimMIM | 0.061 | 0.397 | 0.642 | 0.847 | 0.078 | 0.616 | 0.849 | 0.967 | 0.056 | 0.527 | 0.703 | 0.880 | 0.179 | 0.769 | 0.876 | 0.960 | 0.272 | 0.659 | 0.814 | 0.938 | 10.0 | 9.9 | 9.0 | 8.4 |
| SimCLR | 0.063 | 0.060 | 0.806 | 0.893 | 0.066 | 0.075 | 0.936 | 0.976 | 0.063 | 0.800 | 0.897 | 0.943 | 0.231 | 0.882 | 0.939 | 0.961 | 0.781 | 0.918 | 0.948 | 0.968 | 9.8 | 8.8 | 7.4 | 6.2 |
| DINO | 0.068 | 0.073 | 0.798 | 0.911 | 0.084 | 0.070 | 0.062 | 0.687 | 0.055 | 0.745 | 0.870 | 0.926 | 0.210 | 0.898 | 0.929 | 0.969 | 0.792 | 0.922 | 0.946 | 0.968 | 4.4 | 4.8 | 6.2 | 5.6 |
| VICReg | 0.056 | 0.060 | 0.799 | 0.895 | 0.078 | 0.072 | 0.918 | 0.971 | 0.064 | 0.710 | 0.822 | 0.918 | 0.171 | 0.878 | 0.924 | 0.958 | 0.773 | 0.920 | 0.947 | 0.971 | 5.7 | 6.3 | 5.6 | 5.7 |
| MoCo | 0.049 | 0.055 | 0.840 | 0.928 | 0.061 | 0.063 | 0.944 | 0.985 | 0.056 | 0.847 | 0.919 | 0.965 | 0.166 | 0.911 | 0.939 | 0.969 | **0.860** | **0.944** | **0.961** | 0.974 | 6.4 | 4.7 | 2.2 | 2.4 |
| BYOL | 0.061 | 0.071 | **0.853** | **0.929** | 0.056 | 0.069 | **0.952** | **0.987** | 0.056 | 0.829 | 0.915 | 0.954 | 0.231 | 0.905 | **0.957** | **0.974** | 0.828 | **0.944** | **0.961** | **0.975** | 5.2 | 4.1 | **1.7** | **1.9** |
| Tri-ReD | 0.197 | 0.551 | 0.769 | 0.863 | 0.369 | **0.843** | 0.918 | 0.962 | 0.374 | 0.817 | 0.912 | 0.968 | **0.693** | **0.920** | 0.940 | 0.947 | 0.239 | 0.655 | 0.834 | 0.930 | 3.2 | 7.0 | 8.3 | 9.0 |
| SSL-MAE-$w_\ell$ | 0.153 | 0.555 | 0.757 | 0.889 | 0.212 | 0.797 | 0.933 | 0.965 | 0.463 | 0.881 | 0.950 | 0.979 | 0.590 | 0.876 | 0.922 | 0.937 | 0.656 | 0.889 | 0.919 | 0.958 | 3.0 | 4.2 | 5.7 | 6.5 |
| SSL-MAE-GS | **0.208** | **0.577** | 0.778 | 0.879 | **0.386** | 0.835 | 0.933 | 0.960 | **0.501** | **0.882** | **0.968** | **0.988** | 0.673 | 0.894 | 0.936 | 0.955 | 0.661 | 0.895 | 0.944 | 0.969 | **2.0** | **2.6** | 4.4 | 5.8 |

| MLC Datasets | UCM N (%) | | | | AID N (%) | | | | DFC-15 N (%) | | | | BEN-43 N (%) | | | | MLRSNet N (%) | | | | Avg. ranks N (%) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| SL Baseline | 0.201 | 0.333 | 0.410 | 0.944 | 0.320 | 0.450 | 0.592 | 0.747 | 0.344 | 0.760 | 0.873 | 0.919 | 0.072 | 0.151 | 0.211 | 0.266 | 0.587 | 0.833 | 0.888 | 0.923 | 7.8 | 7.6 | 6.6 | 8.0 |
| SimMIM | 0.208 | 0.726 | 0.838 | 0.934 | 0.303 | 0.550 | 0.657 | 0.717 | 0.362 | 0.773 | 0.889 | 0.954 | 0.138 | 0.182 | 0.226 | 0.270 | 0.479 | 0.734 | 0.820 | 0.895 | 7.3 | 7.2 | 6.0 | 8.2 |
| SimCLR | 0.220 | 0.231 | 0.926 | 0.951 | 0.330 | 0.680 | 0.719 | 0.767 | 0.376 | 0.887 | 0.924 | **0.958** | 0.078 | 0.208 | 0.244 | 0.297 | **0.775** | 0.894 | 0.920 | 0.941 | 4.6 | 4.2 | 2.9 | 4.1 |
| DINO | 0.258 | 0.233 | 0.665 | 0.959 | 0.307 | 0.666 | 0.738 | 0.774 | 0.385 | 0.880 | 0.923 | 0.955 | 0.081 | 0.212 | 0.238 | 0.298 | 0.733 | 0.892 | 0.914 | 0.942 | 4.4 | 4.4 | 5.3 | 3.4 |
| VICReg | 0.228 | 0.221 | 0.898 | 0.948 | 0.303 | 0.635 | 0.704 | 0.758 | 0.401 | 0.878 | 0.923 | 0.957 | 0.077 | 0.206 | 0.241 | 0.303 | 0.757 | 0.887 | 0.918 | 0.944 | 4.9 | 6.4 | 7.0 | 3.8 |
| MoCo | 0.256 | 0.261 | 0.926 | 0.955 | 0.304 | **0.701** | **0.757** | 0.786 | 0.382 | 0.883 | **0.931** | 0.955 | 0.075 | 0.209 | 0.241 | 0.294 | 0.746 | **0.898** | **0.921** | **0.946** | 5.6 | 3.0 | **2.0** | 3.1 |
| BYOL | 0.222 | 0.239 | **0.927** | **0.960** | 0.319 | 0.692 | 0.751 | **0.787** | 0.391 | **0.898** | 0.930 | **0.958** | 0.076 | **0.224** | **0.251** | **0.317** | 0.692 | 0.878 | 0.906 | 0.942 | 5.6 | 3.0 | 2.2 | **1.6** |
| Tri-ReD | 0.516 | 0.851 | 0.903 | 0.948 | 0.526 | 0.662 | 0.703 | 0.757 | 0.673 | 0.789 | 0.855 | 0.880 | 0.148 | 0.201 | 0.205 | 0.275 | 0.416 | 0.748 | 0.843 | 0.918 | 3.0 | **2.1** | 3.6 | 2.4 |
| SSL-MAE-$w_\ell$ | 0.390 | 0.809 | 0.888 | 0.941 | 0.506 | 0.656 | 0.714 | 0.755 | 0.633 | 0.861 | 0.919 | 0.953 | 0.150 | 0.204 | 0.208 | 0.286 | 0.655 | 0.830 | 0.891 | 0.918 | 3.0 | 6.0 | 6.8 | 7.3 |
| SSL-MAE-GS | **0.520** | **0.857** | 0.905 | 0.953 | **0.538** | 0.675 | 0.706 | 0.775 | **0.677** | 0.895 | 0.925 | 0.944 | **0.158** | 0.211 | 0.242 | 0.286 | 0.720 | 0.865 | 0.895 | 0.927 | **1.8** | 3.2 | 4.4 | 5.5 |

The results are presented using the AUPRC measure for MCC datasets and the AUPRC measure for MLC datasets (higher values indicate better predictive performance). The last set of columns denotes average ranks across datasets. For each dataset and percentage of labeled data, the best performance is highlighted in bold, while the second best is underlined.

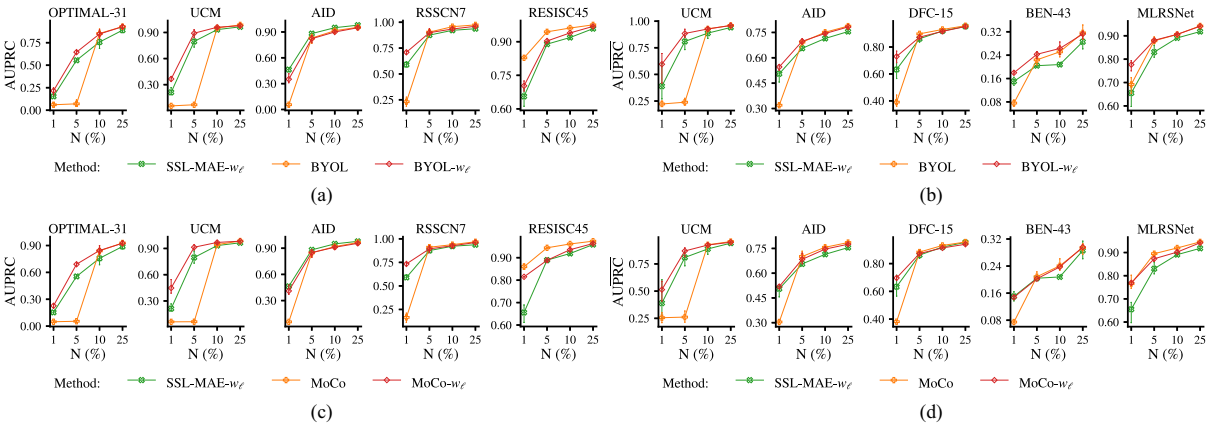



Fig. 5. Comparison of the predictive performance before and after applying the adaptive joint learning process to BYOL (a) and (b) and MoCo (c) and (d). The learning curves depict $\mu \pm \sigma$ (mean and standard deviation) for AUPRC (for MCC) and AUPRC (for MLC) across three repeats, where higher values indicate better predictive performance. Note that the $\sigma$ values are typically very small and are not visible in the graphs (especially at higher percentages of labeled data. (a) MCC datasets (BYOL). (b) MLC datasets (BYOL). (c) MCC datasets (MoCo). (d) MLC datasets (MoCo).

*1) Visualization of the Learned Embeddings:* To visualize the high-dimensional embeddings learned by SSL-MAE, we use the uniform manifold approximation and projection (UMAP) [68]. To quantify the results, we use averaged class-wise variances [69] and Calinski Hurabasz (CH) score [70], also known as the variance ratio criterion. The first one measures the intraclass variations, where a lower value indicates more cohesive and discriminative clusters, while the CH score measures the interclass separability, where a higher value indicates better separability between the different clusters, thus better clustering performance.

The results are depicted in Figs. 7 and 8 for the MCC and MLC versions of the AID dataset, respectively. For AID-MCC, it is evident that the intraclass variations are more preserved in the case of SSL-MAE-GS and SSL-MAE-$w_\ell$ compared to the SL Baseline ($w = 1$). This indicates that the learned embeddings by the joint objective are more efficiently pulled to their corresponding class centers. On the other hand, when solely relying on the unsupervised learning objective ($w = 0$), there is no clear separation of the classes in the embedding space. For the AID-MLC the interclass separability is less pronounced as compared to the MCC case due to the presence of high correlation as well

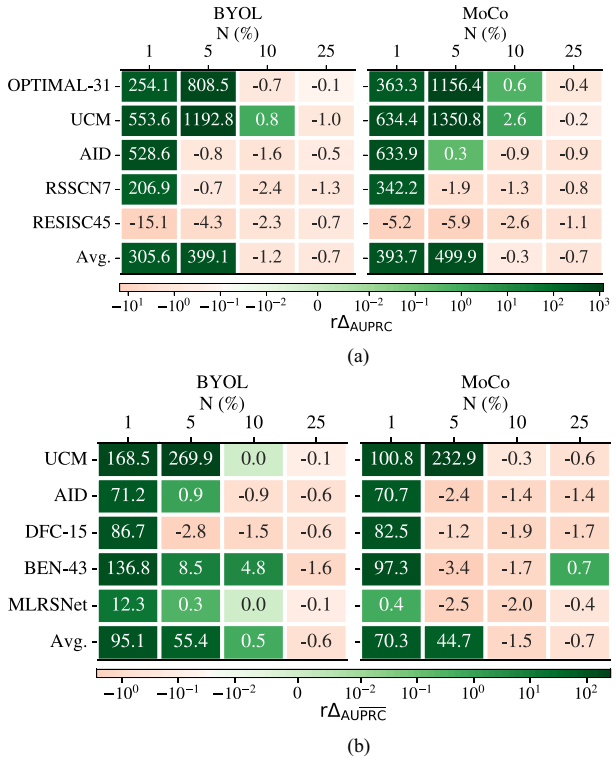



Fig. 6. Relative difference between the performance of the methods before and after applying the adaptive joint learning process. The results highlighted in green indicate improvement in predictive performance and red the opposite. The relative difference is calculated as: $r\Delta_M(\%) = \frac{M_w - M_{w/o}}{M_{w/o}} \cdot 100$. In the case of MCC, $M$ is AUPRC, while for MLC, $M$ is $\overline{\text{AUPRC}}$. (a) MCC datasets. (b) MLC datasets.

as imbalances among the labels in the MLC task. As seen from Fig. 8, both SSL-MAE-GS and SSL-MAE-$w_\ell$, when compared to the SL Baseline, yield better separation between the classes in the embedding space and ensure higher discriminability of the learned feature representations when the data scarcity is more pronounced at smaller percentages of labeled examples. Lastly, the same as in the MCC case, utilizing only the unsupervised learning objective did not produce any separation of the classes in the embedding space, which points to the fact that at least a small amount of supervision is needed to ensure higher quality of the learned representations.

*2) Attention Maps:* Since SSL-MAE utilizes ViT architecture in its encoder part, we can analyze in-depth the model's learning capabilities, primarily due to the self-attention mechanism, a fundamental aspect of the ViT [71]. According to this, we generate attention maps learned from the self-attention heads which helps us gain insights about the model focus on the different parts of the input. Since DeiT-B/16 includes multiple attention heads (12 in this case), we utilize Attention Rollout [72], a technique for mixing attention across tokens through all layers, thus providing a holistic view of the information flow through the self-attention mechanism. For a detailed explanation of the self-attention and the attention rollout, the reader is referred to [71] and [72], respectively.

Figs. 9 and 10 show qualitative results for AID-MCC and AID-MLC datasets, respectively. The results indicate that SSL-MAE-GS and SSL-MAE-$w_\ell$ perform similarly, i.e., the methods attend to image regions that are semantically relevant for classification rather than focusing on the background of the RSI. As opposed to that, when solely relying on the unsupervised term as a learning objective ($w = 0$), the model can not capture the objects present in the RSI properly. Compared to the SL Baseline ($w = 1$), SSL-MAE-GS and SSL-MAE-$w_\ell$ can attend to the image objects more precisely even with fewer labeled examples. This effect can be mostly seen in the AID-MLC dataset, where the complexity of the labels is more pronounced. Lastly, the attention maps also reveal that besides MCC and MLC tasks, SSL-MAE has the potential to address more challenging downstream tasks such as image segmentation.

## VI. Discussion

The analysis of our experimental results offers valuable insights into the effectiveness of SSL-MAE for RS image classification and the broader implications of our adaptive joint learning approach. In this section, we explore the significance of our findings and remaining challenges.

### A. *Effectiveness for RSiM*

Our experimental evaluation demonstrates that SSL-MAE effectively addresses the fundamental challenge of limited labeled data in RS applications. The consistent outperformance of SSL-MAE over supervised baselines across all datasets, with substantial performance gains when using only 1% labeled data (see Figs. 2 and 4), confirms the method's practical utility for EO applications where annotation resources are constrained.

While some established semisupervised approaches occasionally degraded performance compared to supervised learning (see Table III), our method's adaptive joint loss mechanism effectively prevented such degradation by dynamically calibrating the supervision level (see Table II). The performance of SSL-MAE remained robust across diverse datasets with varying characteristics (resolution, geographical coverage, scene complexity), suggesting strong generalization capabilities essential for operational RS applications.

The qualitative evaluations provide deeper insights into SSL-MAE's behavior. The UMAP visualizations reveal that our joint learning objective produces more cohesive class clusters with clearer interclass separation compared to purely supervised or unsupervised approaches (see Figs. 7 and 8). This enhanced feature space organization explains the improved classification performance and demonstrates how combining supervised and self-supervised signals leads to more discriminative representations even with minimal labeled examples.

### B. *Adaptive Joint Learning as a Generalizable Framework*

Perhaps the most significant finding is that our adaptive joint learning strategy extends beyond the specific SSL-MAE architecture. When integrated with state-of-the-art self-supervised methods like BYOL and MoCo, it dramatically enhanced their

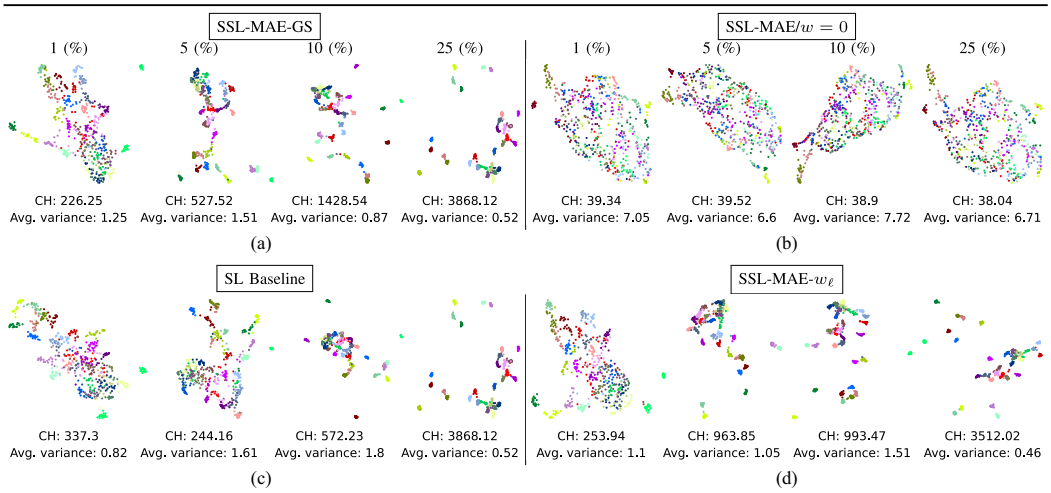



Fig. 7. 2-D UMAP visualization of the learned AID embeddings for the MCC task obtained from different learning settings of SSL-MAE: (a) GS, (b) only using the unsupervised objective $\mathcal{L}_u$ ($w = 0$), (c) using the supervised objective $\mathcal{L}_s$ ($w = 1$), and (d) using the learned weight $w_l$. Different colors represent different classes.

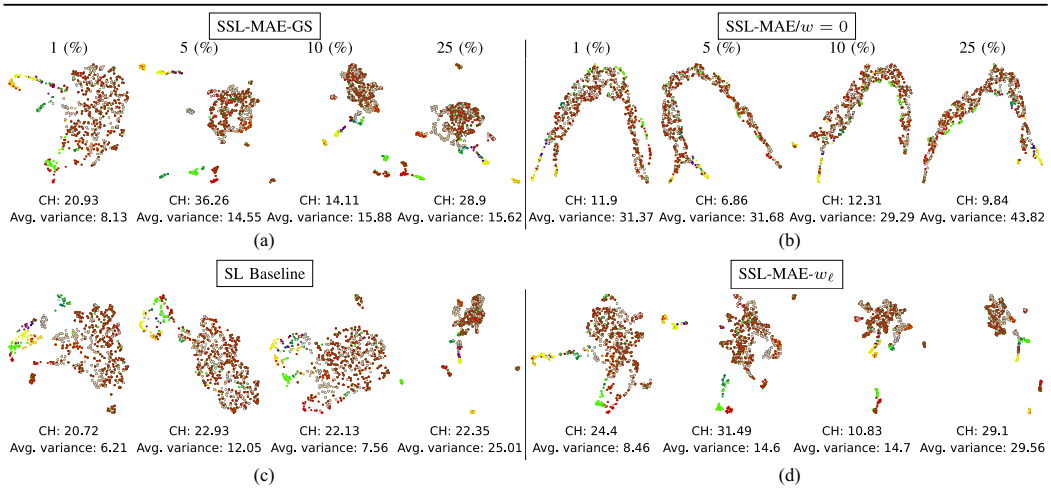



Fig. 8. 2-D UMAP visualization of the learned AID embeddings for the MLC task obtained from different learning settings of SSL-MAE: (a) GS, (b) only using the unsupervised objective $\mathcal{L}_u$ ($w = 0$), (c) only using the supervised objective $\mathcal{L}_s$ ($w = 1$), and (d) using the learned weight $w_l$. Different colors represent different classes.

performance with limited labeled data while preserving their strengths with abundant labels (see Figs. 5 and 6).

This transferability suggests that our approach represents a generalizable framework for improving representation learning under limited supervision. Unlike traditional two-stage pipelines that separate pretraining and fine-tuning, adaptive joint learning creates a continuous spectrum between unsupervised and supervised learning, allowing models to find the optimal balance for each dataset and labeling scenario. The learning of this balance can occur either through GS optimization (SSL-MAE-GS) or through automatic parameter learning (SSL-MAE-$w_\ell$), providing flexibility based on computational constraints.

The attention map analysis further demonstrates how our method directs model focus toward semantically relevant regions, particularly in complex multilabel scenarios (see Figs. 9 and 10). This targeted attention mechanism explains the method's effectiveness in extracting meaningful patterns from limited supervision and suggests potential applications beyond classification, such as weakly-supervised object localization or segmentation in RS imagery.

### C. Challenges and Limitations

Despite its strong performance, SSL-MAE has several limitations that warrant discussion. First, the computational cost of SSL-MAE-GS is significantly higher than supervised baselines due to the GS optimization of the weight parameter. While SSL-MAE-$w_\ell$ offers a favorable performance-speed tradeoff—approximately one order of magnitude faster than SSL-MAE-GS—it still requires more computational resources than purely supervised approaches, which may limit deployment in resource-constrained environments (see Fig. 3).

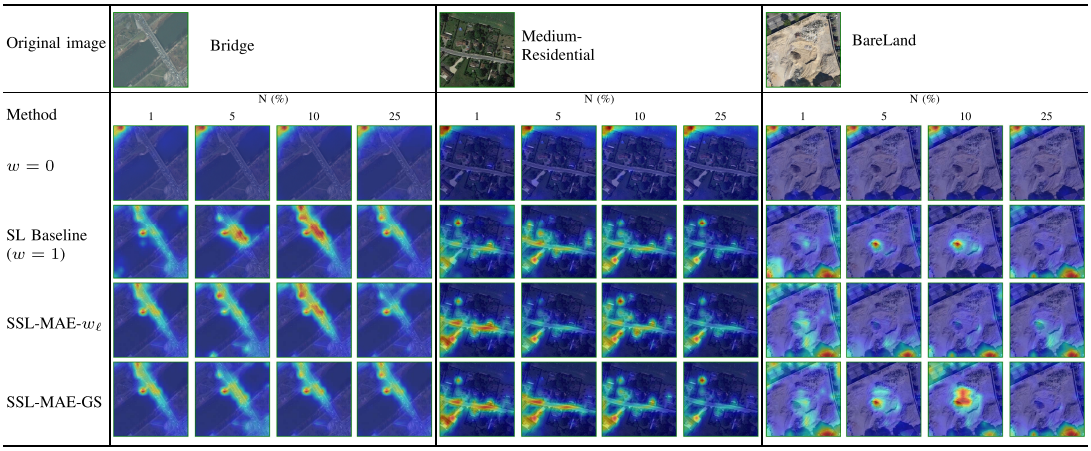



Fig. 9. Self-attention maps obtained from different methods over AID-MCC datasets. The results demonstrate that SSL-MAE-GS leads to more expressive semantics, while relying solely on the unsupervised loss ($w = 0$) prevents the encoder from capturing the semantic context in RSIs properly.

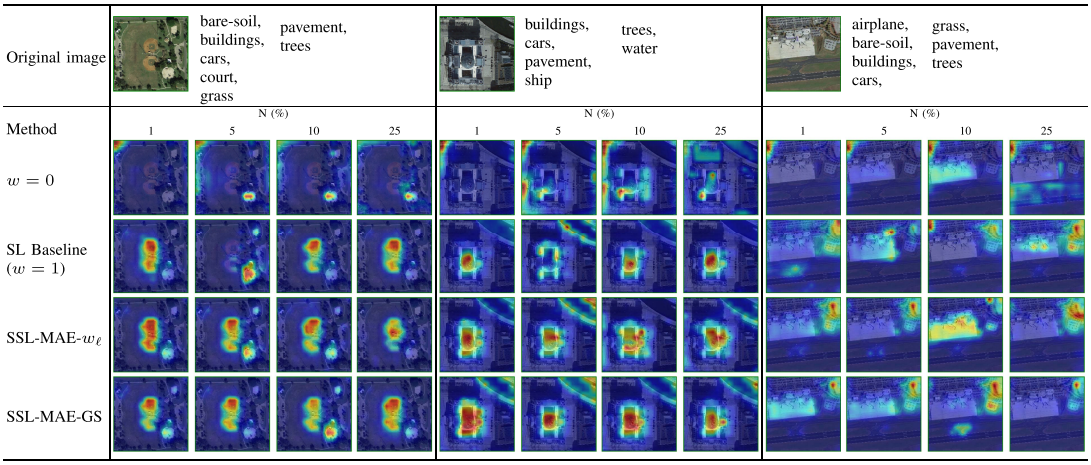



Fig. 10. Self-attention maps obtained from different methods over AID-MLC datasets. The results demonstrate that SSL-MAE-GS leads to more expressive semantics, while when relying solely on the unsupervised loss ($w = 0$), the encoder cannot capture the semantic context in the RSIs properly.

Second, our method does not consistently outperform specialized self-supervised methods like BYOL and MoCo when labeled data exceeds 10% (see Table IV). This suggests room for improvement in scaling to higher supervision regimes while maintaining advantages in low-data scenarios. The crossover point where self-supervised methods begin to outperform SSL-MAE varies by dataset, indicating that domain-specific factors influence effectiveness.

Third, the performance gap between SSL-MAE and other methods varies across datasets, suggesting that data characteristics affect the relative advantage of our approach. Further investigation into these factors would help predict, which RS applications would benefit most from adaptive joint learning and allow for application-specific optimizations.

## VII. Conclusion

In this study, we present SSL-MAE, a novel end-to-end method for semisupervised multilabel and MCC of RSIs. It consists of a self-supervised branch that leverages the information from the unlabeled data through an MAE model that uses MIM as a pretext task for representation learning, and a supervised branch involving a discriminative classifier to obtain the necessary supervision from the limited amount of labeled data. We showed that the learned features can be strongly enhanced by the joint learning process that combines the information from the supervised and the self-supervised branches through an adaptive joint loss function that controls the amount of supervision.

Through rigorous and extensive experiments, we showed that SSL-MAE achieves excellent results in the challenging semisupervised setting, consistently outperforming the supervised baseline and successfully exploiting unlabeled data. We also compared our approach against several state-of-the-art semisupervised and self-supervised methods from the literature, showing that SSL-MAE consistently outperforms the semisupervised methods while achieving competitive performance against the two-stage self-supervised methods in scenarios with small amounts of labeled data. In addition, our findings suggest

that the successful application of the adaptive joint learning process is not restricted to the specific design we proposed, having the potential to improve a broad spectrum of self-supervised techniques. Specifically, the integration of adaptive joint learning with two most effective self-supervised methods (BYOL and MoCo), greatly enhanced their capabilities in dealing with labeled data scarcity.

Given the promising results, further extension of this work will focus on several different aspects, such as careful exploitation of the unlabeled data, exploring the learning capabilities in a few-shot setting, as well as addressing other more challenging downstream tasks such as image segmentation.

## REFERENCES


[1] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, "Remote sensing image classification: A comprehensive review and applications," *Math. Problems Eng.*, vol. 2022, pp. 1–24, Aug. 2022. [Online]. Available: https://www.hindawi.com/journals/mpe/2022/5880959/

[2] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *WIREs Data Mining Knowl. Discov.*, vol. 8, no. 6, Nov. 2018, Art. no. e1264.

[3] W. Han et al., "A survey of machine learning and deep learning in remote sensing of geological environment: Challenges, advances, and opportunities," *ISPRS J. Photogrammetry Remote Sens.*, vol. 202, pp. 87–113, 2023.

[4] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, and M. Ryo, "Ten deep learning techniques to address small data problems with remote sensing," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 125, 2023, Art. no. 103569.

[5] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 4, pp. 213–247, Dec. 2022.

[6] M. Oquab et al., "Dinov2: Learning robust visual features without supervision," 2023, *arXiv:2304.07193*.

[7] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8934–8954, Sep. 2022.

[8] K. Sohn et al., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 596–608.

[9] B. Zhang et al., "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 18408–18419.

[10] Y. Oh, D.-J. Kim, and I. S. Kweon, "DASO: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9786–9796.

[11] W. Miao, J. Geng, and W. Jiang, "Semi-supervised remote-sensing image scene classification using representation consistency siamese network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5616614.

[12] P. Gómez and G. Meoni, "MSMatch: Semisupervised multispectral scene classification with few labels," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11643–11654, 2021.

[13] J. Kang, R. Fernández-Beltrán, Z. Ye, X. Tong, P. Ghamisi, and A. Plaza, "High-rankness regularized semi-supervised deep metric learning for remote sensing imagery," *Remote Sens.*, vol. 12, no. 16, 2020, Art. no. 2603.

[14] M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Semi-supervised multi-label classification of land use/land cover in remote sensing images with predictive clustering trees and ensembles," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4706416.

[15] L. Wang, Y. Liu, H. Di, C. Qin, G. Sun, and Y. Fu, "Semi-supervised dual relation learning for multi-label classification," *IEEE Trans. Image Process.*, vol. 30, pp. 9125–9135, 2021.

[16] H. Cevikalp, B. Benligiray, Ö. N. Gerek, and H. Saribas, "Semi-supervised robust deep neural networks for multi-label classification." in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 9–17.

[17] H. Cevikalp, B. Benligiray, and O. N. Gerek, "Semi-supervised robust deep neural networks for multi-label image classification," *Pattern Recognit.*, vol. 100, 2020, Art. no. 107164.

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 16000–16009.

[19] Z. Xie et al., "SimMIM: A simple framework for masked image modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 9653–9663.

[20] M. Singh et al., "The effectiveness of mae pre-pretraining for billion-scale pretraining," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 5484–5494.

[21] S. Atito, M. Awais, A. Farooq, Z. Feng, and J. Kittler, "MC-SSL0. 0: Towards multi-concept self-supervised learning," 2021, *arXiv:2111.15340*.

[22] J. Xiao, Y. Bai, A. Yuille, and Z. Zhou, "Delving into masked autoencoders for multi-label thorax disease classification," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2023, pp. 3588–3600.

[23] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[24] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 9650–9660.

[25] F. Liang, Y. Li, and D. Marculescu, "SupMAE: Supervised masked autoencoders are efficient vision learners," 2022, *arXiv:2205.14540*.

[26] Y. Cong et al., "SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2022, pp. 197–211.

[27] J. Lin, F. Gao, X. Shi, J. Dong, and Q. Du, "SS-MAE: Spatial–spectral masked autoencoder for multisource remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5531614.

[28] X. Cao, H. Lin, S. Guo, T. Xiong, and L. Jiao, "Transformer-based masked autoencoder with contrastive loss for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5524312.

[29] X. Sun et al., "RingMO: A remote sensing foundation model with masked image modeling," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5612822.

[30] Y. Bai et al., "Masked autoencoders enable efficient knowledge distillers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2023, pp. 24256–24265.

[31] J. Jakubik et al., "Foundation models for generalist geospatial artificial intelligence," Oct. 2023, *arXiv:2310.18660*.

[32] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[33] Y. Wang et al., "Freematch: Self-adaptive thresholding for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2023, pp. 1–20.

[34] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10687–10698.

[35] I. Nassar, S. Herath, E. Abbasnejad, W. Buntine, and G. Haffari, "All labels are not created equal: Enhancing semi-supervision via label grouping and co-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7241–7250.

[36] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 142–159.

[37] M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Semi-supervised multi-label classification of land use/land cover in remote sensing images with predictive clustering trees and ensembles," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4706416.

[38] Z. Zhao, Z. Luo, J. Li, C. Chen, and Y. Piao, "When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework," *Remote Sens.*, vol. 12, no. 20, 2020, Art. no. 3276.

[39] H. Liu, M. Zhang, Z. Di, M. Gong, T. Gao, and A. K. Qin, "A hybrid multitask learning network for hyperspectral image classification with few labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5505816.

[40] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[41] R. Xiong et al., "On layer normalization in the transformer architecture," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, vol. 119, pp. 10524–10533.

[42] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019.

[43] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th ACM SIGSPATIAL Int. Symp. Adv. Geographic Inf. Syst.*, 01 2010, pp. 270–279.

[44] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[45] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[46] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
[47] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1144–1158, Feb. 2018.
[48] Y. Hua, L. Mou, and X. Zhu, "Relation network for multi-label aerial image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4558–4572, Jul. 2020.
[49] Y. Hua, L. Mou, and X. Zhu, "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 149, pp. 188–199, 2019.
[50] Q. X. Qi et al., "MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding," *ISPRS J. Photogrammetry Remote Sens.*, vol. 169, pp. 337–350, 2020.
[51] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "BigEarthNet: A large-scale benchmark archive for remote sensing image understanding," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, vol. 12, no. 2, pp. 5901–5904.
[52] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, vol. 139, pp. 10347–10357.
[53] J. Deng, W. Dong, R. Socher, L. Li, L. Kai, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[54] E. Fini et al., "Semi-supervised learning made simple with self-supervised clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3187–3197.
[55] Y. Wang et al., "USB: A unified semi-supervised learning benchmark for classification," in *Proc. 36th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2022, pp. 3938–3961.
[56] J. Li, C. Xiong, and S. C. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9475–9484.
[57] H. Chen et al., "Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning," 2023, *arXiv:2301.10921*.
[58] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "Simmatch: Semi-supervised learning with similarity matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14471–14481.
[59] I. Susmelj et al., "Lightly," GitHub. Note, 2020. [Online]. Available: https://github.com/lightly-ai/lightly
[60] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
[61] J. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020. [Online]. Available: https://arxiv.org/abs/2006.07733
[62] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–23.
[63] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019. [Online]. Available: http://arxiv.org/abs/1911.05722
[64] K. Zhao, J. Li, S. Xie, L. Zhou, W. He, and X. Chen, "Self-supervised learning with trilateral redundancy reduction for urban functional zone identification using street-view imagery," *Sensors*, vol. 25, no. 5, 2025, Art. no. 1504. [Online]. Available: https://www.mdpi.com/1424-8220/25/5/1504
[65] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[66] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning* (Adaptive Computation and Machine Learning). Cambridge, MA, USA: MIT Press, 2006.
[67] Y.-F. Li and D.-M. Liang, "Safe semi-supervised learning: A brief introduction," *Front. Comput. Sci.*, vol. 13, pp. 669–676, 2019.
[68] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, 2018, Art. no. 861.
[69] Y. Liu, H. Li, C. Hu, S. Luo, Y. Luo, and C. W. Chen, "Learning to aggregate multi-scale context for instance segmentation in remote sensing images," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 595–609, Jan. 2025.
[70] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.*, vol. 3, no. 1, pp. 1–27, 1974.
[71] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[72] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 4190–4197.



**Marjan Stoimchev** received the M.S. degree in biometrics from the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia, in 2021. He is currently working toward the Ph.D. degree in information and communication technologies, computer science with the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia.

His main research focus lies in computer vision, specifically in deep learning, classical machine learning, and the combination of both. His research interests are also in the field of multilabel classification of remote sensing images with supervised and semisupervised learning.



**Jurica Levatić** received the Ph.D. degree from the JSI Postgraduate School, Ljubljana, Slovenia, in 2017. His Ph.D. dissertation was titled "Semisupervised Learning for Structured Output Prediction."

He is currently a Researcher with the Department of Knowledge Technologies, JSI. From 2018 to 2021, he was a Marie Skłodowska-Curie Postdoctoral Fellow with the Institute for Research in Biomedicine, Barcelona, Spain. His research interests are focused in cancer biology semisupervised learning, and application of machine learning to solve practically relevant problems.



**Dragi Kocev** received the Ph.D. degree from the JSI Postgraduate School, Ljubljana, Slovenia, in 2011. He Ph.D. dissertation was titled "Learning Ensemble Models for Predicting Structured Outputs."

He is currently a Researcher with the Department of Knowledge Technologies, JSI. He was a Visiting Research Fellow with the University of Bari, Bari, Italy, in 2014/2015. He was involved with the EU funded projects TRUSTroke, IQ, PHAGOSYS, and HBP. He was a Co-Coordinator of the FP7 FET Open project MAESTRA and coordinator of ESA funded activities AiTLAS and GalaxAI.

Dr. Kocev is member of the PC for several conferences (incl. DS, European Conference on Machine Learning (ECML), Principles and Practice of Knowledge Discovery in Databases (PKDD), Association for the Advancement of Artificial Intelligence, International Joint Conference on Artificial Intelligence, Knowledge Discovery and Data Mining) and action editor for Expert Systems with *Applications*, *Machine learning*, and *Data Mining and Knowledge Discovery*. He was a PC Co-Chair for DS 2014 and Journal track co-chair for ECML PKDD 2017.



**Sašo Džeroski** is currently a Head of the Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia. His department develops artificial intelligence methods for machine learning and decision support and uses them to solve practical problems from agriculture and environmental sciences, medicine and life sciences, and space operations/Earth observation. His own research focuses on artificial intelligence for science, including machine learning from complex data and in the presence of domain knowledge, learning models of dynamical systems, and developing ontologies for describing different branches of computer science research. He is currently the Vice President of the Slovenian Artificial Intelligence Society.

Mr. Džeroski was a fellow of EurAI, the European Association of Artificial Intelligence. He was the recipient of "Pioneering Work in AI and Outstanding Service for the European AI community." He is currently a member of the Macedonian Academy of Sciences and Arts and a member of Academia Europea (European Academy of Humanities, Letters and Sciences).

# Chapter 6

# Hierarchical Multi-Label Classification of Remote Sensing Images With Supervised and Semi-Supervised Learning

This chapter presents our contributions to hierarchical multi-label classification (HMLC) of remote sensing images (RSIs), addressing the limitations of existing methods in modeling complex, multi-path hierarchies and leveraging unlabeled data. It contains two complementary studies. The first introduces hierarchical multi-label classification of land use and land cover in RSIs, demonstrating the benefits of modeling hierarchical dependencies. The second extends this work with the *Hierarchical and Explicit Label Modeling* (HELM) framework, which unifies hierarchical graph learning and semi-supervised learning (SSL) for HMLC.

From the hypotheses defined in Section 1.3, this chapter addresses the following:

**H5** Incorporating hierarchical information in multi-label image classification improves predictive performance by capturing structured label dependencies. Extending this with graph-based reasoning and semi-supervised learning further enhances generalization and robustness across domains.

Our results confirm **H5** across both studies. In the first, we show that hierarchical information improves predictive performance in RSI classification, particularly under limited labeled data. In the second, we extend this idea with HELM to eight datasets across remote sensing and medical imaging domains, demonstrating improved generalization and robustness. HELM consistently outperforms both MLC and HMLC baselines in supervised and semi-supervised settings.

## 6.1 Hierarchical Multi-Label Classification of Land Use/Land Cover in Remote Sensing Images

The first study investigates whether explicit use of label hierarchies improves the predictive performance in RSI classification. By aligning the label taxonomies of BigEarthNet, UCM, and AID with the CORINE Land Cover (CLC) nomenclature, we construct hierarchical datasets that enable systematic analysis of hierarchy-aware learning effects on predictive accuracy. In doing so, we introduce and release the first HMLC benchmarks for remote sensing, extending the BigEarthNet dataset (in both its 19-class and 43-class CLC-aligned

versions) as well as the UCM and AID datasets with multi-level label hierarchies derived from the CLC taxonomy. These resources constitute the first publicly available benchmarks for evaluating hierarchical classification in remote sensing. The complete datasets and label structures are available online at: `https://huggingface.co/marjandl`. The experiments show that incorporating hierarchical dependencies yields measurable improvements, especially when labeled data are scarce.

The paper included in this section is:

- M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Hierarchical multi-label classification of land use/land cover in remote sensing images", in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Revised Selected Papers, Part III*, *Communications in Computer and Information Science*, vol. 2560, in press, Cham: Springer, 2026.

**The contributions of Marjan Stoimchev to this paper are as follows.** Marjan Stoimchev constructed the hierarchical label mappings for the BigEarthNet, UCM, and AID datasets, implemented and executed the experiments, performed the analyses, and prepared and revised the manuscript in collaboration with the co-authors.

**Code availability.** All code developed for this study, including model implementations, training pipelines, evaluation scripts, and instructions for loading the hierarchical datasets, is publicly available at: `https://github.com/marjanstoimchev/hmlc_net`.

# Hierarchical Multi-label Classification of Land Use/ Land Cover in Remote Sensing Images

Marjan Stoimchev[1,2] *, Jurica Levatić[1], Dragi Kocev[1], and Sašo Džeroski[1]

[1] Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia
{marjan.stoimchev}@ijs.si

**Abstract.** The task of hierarchical multi-label classification (HMLC) involves predicting structured outputs, where classes are organized in a hierarchy and instances can be associated with multiple classes simultaneously. Despite the popularity of HMLC in the machine learning field, there is a lack of HMLC datasets and methods in the remote sensing (RS) domain. Therefore, it is still not clear whether considering the hierarchical relationships among classes of remote sensing images (RSI) is advantageous or not. The CORINE Land Cover (CLC) nomenclature, which provides detailed information about the land cover classes at multiple levels of the hierarchy, makes it possible to construct label hierarchies for RSI datasets and explore the relevance of HMLC in the RS domain. In this work, based on the CLC nomenclature, we assemble label hierarchies for the BigEarthNet and UCM datasets, constructing HMLC version of these datasets, which we release to the public as the first HMLC RSI benchmarks. Furthermore, we show that incorporating hierarchical information about the classes brings advantage in predictive performance over the prevalent multi-label classification setting.

**Keywords:** Remote sensing · Multi-label classification · Hierarchical multi-label classification.

## 1 Introduction

The increasing interest in Earth Observation (EO) has led to significant technological progress, resulting in the creation of large-scale remote sensing image (RSI) archives [1]. This has attracted a considerable amount of research in various application areas, particularly in employing machine learning (ML) and deep learning (DL) methods for analysis [2]. Two of the most common ML tasks in RSI analysis are multi-class classification (MCC) [3] and multi-label classification (MLC) [4]. In MCC, the objective is to assign a single label or semantic category to an image, which often oversimplifies the classification process. Conversely, addressing the RSI classification task through MLC enables a more realistic representation of the complex content within the images [5].

Many real-life predictive modeling problems involve structured outputs, where dependencies exist between classes. These dependencies can manifest in various

forms, such as classes organized into a tree-shaped hierarchy or a directed acyclic graph, or through internal relations between classes, such as sequences. The task of hierarchical multi-label classification (HMLC) deals with problems where each sample can be associated with multiple hierarchically organized labels. HMLC has been successfully applied in diverse fields of applications, such as predicting gene function [6], categorizing and analyzing text [7], analysis of diseases in medical images [8], classification of species [9], and many others. It has been shown that the inclusion of hierarchical information often helps to improve the MLC models [10]. Nowadays, in the computer vision (CV) domain, many existing approaches address the HMLC task by transforming the MCC problem into a MLC problem [11–14]. However, these approaches predict a single path from a predefined hierarchy or taxonomy, ensuring that a single label is assigned at the bottom level of the hierarchy. This has also been applied in the RSI domain, where a RSI is assigned to a specific land-use type within the hierarchy [15, 16]. Such an approach does not fully incorporate the hierarchical structure, thus deviating from real-life applications. To the best of our knowledge, no previous studies in RS address the task where each RSI can be associated with several different paths of the class hierarchy, i.e., multiple labels can coexist within the leaf nodes.

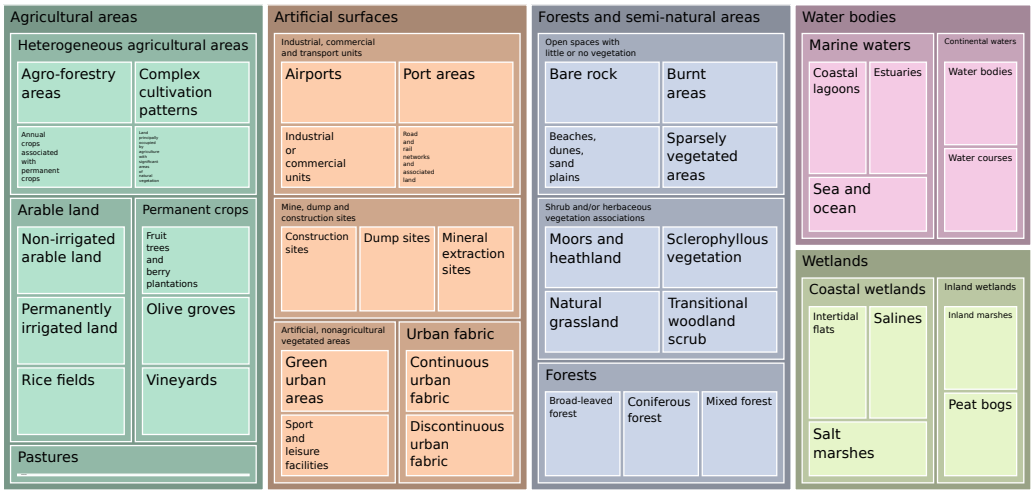
In this paper, we present two HMLC RSI datasets, extensions of the existing BigEarthNet and UCM datasets. Both datasets feature expanded label sets, comprised of labels organized as hierarchies according to the CORINE Land Cover database of the year 2018 (CLC 2018)[3]. We also explore whether the inclusion of hierarchical information in these datasets helps in achieving better predictive performance than achieved with their original MLC versions. To this end, we use a simple HMLC approach that uses hierarchical information within the learning process. We evaluate this approach through extensive experiments against the MLC baseline. The results demonstrate that models learned in a HMLC setting tend to produce better results for small amounts of data.

## 2 Description of HMLC datasets

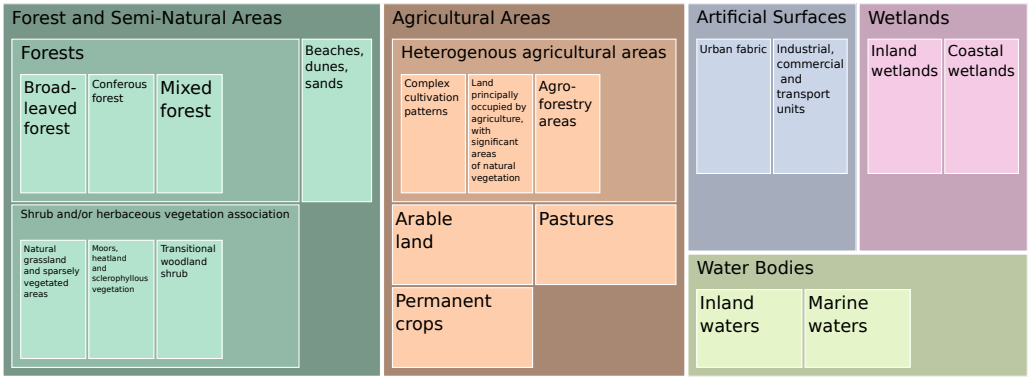
In this work, we construct label hierarchies for the two RSI multi-label datasets, BigEarthNet and UCM.

The BigEarthNet dataset was developed by the Remote Sensing Image Analysis (RSiM) Group and the Database Systems and Information Management (DIMA) Group at the Technische Universität Berlin (TU Berlin) [1]. The dataset consists of 125 Sentinel-2 tiles captured from June 2017 to May 2018 across 10 European countries: Austria, Belgium, Finland, Ireland, Kosovo, Lithuania, Luxembourg, Portugal, Serbia, and Switzerland. The tiles were chosen based on cloud coverage of less than 1%, where each one is associated with 12 spectral bands. The tiles were also divided into 590,326 non-overlapping image patches of 120x120 pixels, where each image patch is annotated with multiple land-cover classes, i.e., multiple labels, taken from the CLC 2018.

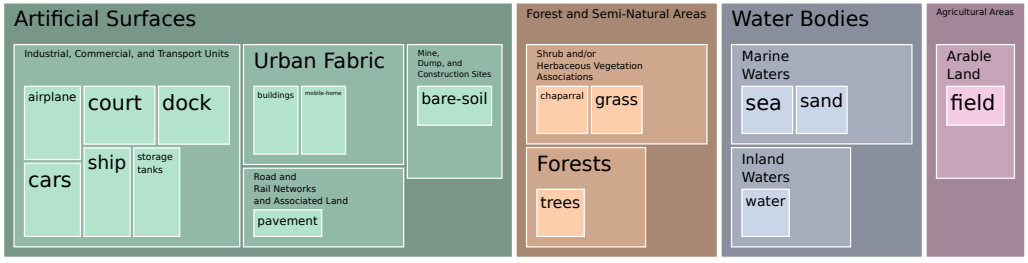
[3] Available online at: https://land.copernicus.eu/en/technical-library




(a) BigEarthNet-43




(b) BigEarthNet-19




(c) UCM

Fig. 1: Hierarchical land cover classes for BigEarthNet, (a) original CLC nomenclature with 43 labels, (b) simplified version of the original hierarchy with 19 labels in the leaves, and (c) for the UCM dataset.

We constructed the label-hierarchy for BigEarthNet on the basis of the CLC nomenclature. The CLC nomenclature includes land cover classes grouped in

a three-level hierarchy, which consists of 43 land-use types at the third (bottom) level, 15 at the second level, and 5 on the first level, where the number of images for each land cover class varies significantly. Later, Sumbul et al. [17] presented an alternative nomenclature of reduced size with 19 land-use types at the third (bottom) level, defined by interpreting and arranging the CLC level-3 nomenclature based on the properties of Sentinel-2 images. In a similar way, we constructed a label-hierarchy for the UCM Land Use dataset, which contains 2,100 images grouped into 21 broad categories at the scene level, with a total of 17 object-level labels [18]. The extended version of the UCM dataset contains 30 labels. The representations of label-hierarchies for BigEarthNet and UCM datasets are shown in Figure 1. The label-hierarchies we constructed for these datasets are available for download[4].

## 3 Methodology

In this section, we propose a simple end-to-end hierarchical multi-label classification approach for categorizing land use and land cover from remote sensing images (Fig. 2). Our method involves predicting class labels at multiple hierarchical levels (Fig. 2, top).

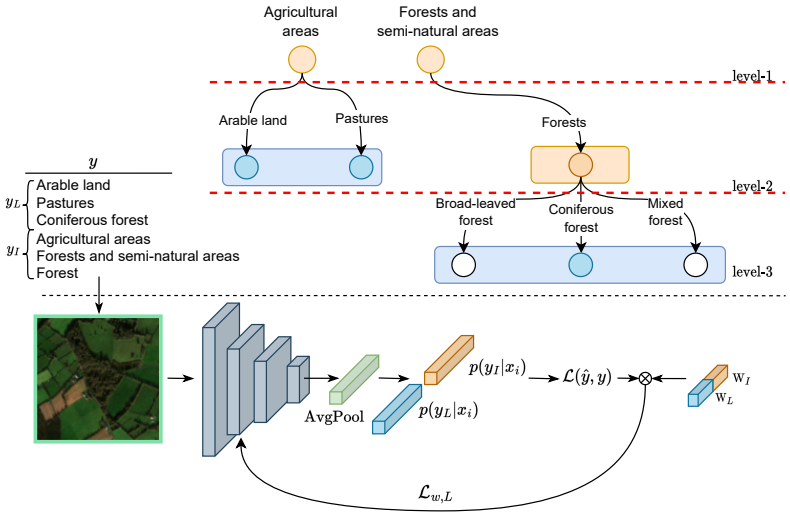



Fig. 2: Overview of the proposed framework for HMLC of RSIs.

First, an input RGB RSI $\mathbf{x}_i \in \mathbb{R}^{w \times h \times 3}$, with width $w$ and height $h$, is processed through a feature extraction component responsible for capturing the rich semantics present in a RSI (Fig. 2, bottom). This component encodes the input

[4] https://huggingface.co/datasets/marjandl/BigEarthNet-43-HMLC, https://huggingface.co/datasets/marjandl/BigEarthNet-19-HMLC, https://huggingface.co/datasets/marjandl/UCM-HMLC

image into a $d$-dimensional feature space $\mathbf{f} \in \mathbb{R}^d$ by using an encoder structure denoted as $E_\theta$, where $\theta$ represents the parameters learned during training. The learned representations are aggregated through an average pooling operation and passed through a single fully connected (FC) layer. This layer comprises a number of output units equal to the total number of leaf and intermediate nodes in the hierarchy. This setup transforms the HMLC problem using a "flattening" approach, consolidating all classes within the hierarchy and treating the problem as a multi-label classification problem. The output from this layer are non-normalized *log* probabilities (*logits*), further converted to a vector of probabilities $p_i$ by using the *sigmoid* activation function $p_i = \sigma(z_i) = 1/(1 + e^{-z_i})$, where $z_i \in \mathbb{R}^c$ is the *logit* vector for the $i$-th sample, and $c$ denotes the total number of classes (both leaf and intermediate). This vector is comprised of the vector of probabilities for the leaf labels $p(y_L \mid x_i) = [\sigma(z_{i1}), \sigma(z_{i2}), \ldots, \sigma(z_{in})]$, and vector of probabilities for the intermediate labels $p(y_I \mid x_i) = [\sigma(z_{i,n+1}), \sigma(z_{i,n+2}), \ldots, \sigma(z_{i,c})]$, where the leaf and the intermediate labels correspond to indices $1, 2, \ldots, n$ and $n+1, n+2, \ldots, c$, respectively.

During training, given a mini-batch of RSIs, the model is optimized with a weighted variant of the binary cross-entropy learning objective (BCE) $\mathcal{L}_{wbce}$, as follows:

$$\mathcal{L}_{wbce} = -\frac{1}{c}\sum_{i=1}^{c}[y_i log(p_i) + (1 - y_i) log(1 - p_i)] \times w_i, \tag{1}$$

where $y_i$ is a binary vector representing the multiple labels from the hierarchy and their corresponding class weights $w$. The weight of a class $c$ decreases exponentially with the depth of the class in the hierarchy, defined as follows:

$$w = w_0^{depth(c)} \tag{2}$$

where $0 < w_0 \leq 1$ is weight at the root level of hierarchy and $depth(c)$ returns an integer denoting the depth of class $c$ in the label-hierarchy.

## 4 Experimental design

To explore how the predictive performance of the learning strategies is affected by data availability, we employ a sub-sampling strategy. From the total number of 590,326 images of BigEarthNet dataset, we set aside a test set of 10,000 images, while from the remaining images we sample fractions of 100, 500, 1000, 2500, 5000, and 10000 images (we also utilize the full set of images). For the UCM dataset, we use 400 images as a test set, while from the remaining, we sample sets of 100, 250, 500, and 1000 images. For each subset, we reserve 10% of the training data for validation purposes. All the splits are done by adopting an iterative stratified sampling strategy to preserve the relative frequency of the labels in the datasets to the maximum possible extent.

We repeated the experiments three times, using different random initializations for each fraction of sampled data. The final predictive performance is averaged across the three runs. Since MLC lacks information about the intermediate

hierarchy, a fair comparison between HMLC and MLC is essential. To this end, we evaluate the HMLC methods based on the predictions of the leaf labels in the hierarchy, which exactly correspond to the MLC labels. For the HMLC methods, we also explore different values of $w_0 \in (0.5, 0.75, 1)$ to vary the importance of the labels at different levels of the hierarchy within the learning objective.

Results are presented in terms of ranking loss ($rl$) and average AUPRC ($\overline{\text{AUPRC}}$), as these measures are independent of classification thresholds and provide a reliable evaluation of the performance. Lower values of ($rl$) and higher values of $\overline{\text{AUPRC}}$ indicate better predictive performance, respectively.

We used two backbone network architectures implemented within the `PyTorch` framework, namely, the base variants of ConvNeXt V2[5] (ConvNeXt V2-B)[19], initialized using ImageNet-1K weights, and the DINOv2, which uses the ViT-B/14 backbone (DINOv2-B) trained on the LVD-142M dataset [20]. The models are fine-tuned for 20 epochs on 4 NVIDIA A100 (40GB) GPUs using the AdamW optimizer, using an initial learning rate of $1 \times 10^{-4}$ and a *cosine* learning rate scheduler, a batch size of 4, and mixed precision of training to speed up the training process. The models are fine-tuned on the BigEarthNet and UCM datasets using $224 \times 224$ resolution RGB images. We also apply strong augmentations to stimulate the model training and prevent overfitting, especially at low data regimes.

## 5 Results and Discussion

The results of the evaluation are examined in the context of the following research questions: (*i*) Does considering the label-hierarchy bring advantage to predictive performance of HMLC as compared to MLC? (*ii*) What impact does data availability have on the predictive performance of MLC and HMLC methods? (*iii*) How effective is the transfer learning performance of MLC and HMLC models when transitioning from a large-scale dataset to a limited-size dataset in both MLC and HMLC scenarios?

### 5.1 Comparison of HMLC and MLC

In the first experiment, we discuss the principal question of this study: Is the label hierarchy beneficial for predictive performance when classifying RSI? We also comment on the impact of data availability on the performance of MLC and HMLC.

The results for BigEarthNet and UCM datasets are presented in Tables 1 and 2, respectively. Overall, we observe that the HMLC method achieves better predictive performance in many cases for both of the datasets, demonstrating the benefits of using the hierarchical information for model training.

[5] We use the Fully Convolutional Masked AutoEncoder (FCMAE) self-supervised pretrained version of ConvNeXt V2-B on ImageNet-22K

Table 1: Comparison between the predictive performance of two network architectures trained on the BigEarthNet-43 and BigEarthNet-19 datasets for both MLC na HMLC tasks. For each subset of data of size $N$, the best performance (learning HMLC with different $w_0$ and MLC) is highlighted in bold.

| $N$ | Task | $w_0$ | BigEarthNet-43 | | | | BigEarthNet-19 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ConvNeXt V2 | | DINOv2 | | ConvNeXt V2 | | DINOv2 | |
| | | | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ |
| 100 | HMLC | 0.5 | **0.125** | **0.105** | 0.121 | 0.080 | **0.168** | **0.267** | **0.206** | **0.223** |
| | | 0.75 | 0.141 | 0.095 | 0.107 | **0.086** | 0.190 | 0.246 | 0.216 | 0.205 |
| | | 1.0 | 0.182 | 0.083 | **0.104** | **0.086** | 0.174 | 0.263 | 0.225 | 0.200 |
| | MLC | - | 0.155 | 0.097 | 0.136 | 0.084 | 0.182 | 0.255 | 0.222 | 0.199 |
| 500 | HMLC | 0.5 | 0.080 | 0.146 | 0.086 | 0.105 | **0.125** | 0.362 | 0.138 | 0.322 |
| | | 0.75 | 0.079 | 0.147 | **0.080** | 0.119 | 0.133 | **0.370** | **0.131** | **0.341** |
| | | 1.0 | 0.081 | 0.144 | 0.089 | 0.108 | 0.136 | 0.341 | 0.135 | 0.328 |
| | MLC | - | **0.077** | **0.149** | 0.086 | **0.133** | **0.125** | 0.368 | 0.137 | 0.331 |
| 1,000 | HMLC | 0.5 | **0.058** | **0.208** | 0.062 | 0.176 | 0.094 | 0.473 | **0.151** | **0.295** |
| | | 0.75 | 0.060 | 0.202 | **0.056** | 0.191 | **0.093** | 0.473 | 0.185 | 0.268 |
| | | 1.0 | 0.061 | 0.202 | 0.060 | 0.178 | 0.094 | **0.482** | 0.183 | 0.282 |
| | MLC | - | 0.059 | 0.207 | 0.059 | **0.204** | 0.094 | 0.472 | 0.160 | 0.292 |
| 10,000 | HMLC | 0.5 | 0.037 | 0.300 | 0.027 | 0.394 | 0.058 | 0.584 | 0.092 | 0.483 |
| | | 0.75 | 0.036 | 0.301 | **0.026** | 0.409 | 0.057 | 0.580 | 0.071 | 0.579 |
| | | 1.0 | 0.036 | 0.299 | **0.026** | **0.428** | **0.056** | 0.586 | **0.067** | **0.600** |
| | MLC | - | **0.034** | **0.337** | 0.039 | 0.332 | **0.056** | **0.616** | 0.092 | 0.460 |
| 580,326 | HMLC | 0.5 | 0.010 | 0.737 | 0.009 | 0.793 | 0.021 | 0.833 | 0.018 | 0.897 |
| | | 0.75 | **0.009** | 0.727 | **0.007** | 0.817 | 0.020 | 0.836 | **0.016** | **0.906** |
| | | 1.0 | 0.010 | 0.732 | 0.008 | 0.821 | 0.020 | 0.834 | 0.017 | 0.904 |
| | MLC | - | 0.010 | **0.767** | 0.008 | **0.846** | **0.019** | **0.870** | 0.017 | 0.903 |

More specifically, for the BigEarthNet dataset and smaller label-hierarchy (BigEarthNet-19), we observe that DINOv2 with HMLC achieves consistent improvements over MLC both in terms of ranking loss and $\overline{\text{AUPRC}}$. For ConvNeXt V2 architecture and the BigEarthNet-43 hierarchy, the results are not so conclusive, even if HMLC tends to outperform MLC more often than not. Similarly, for BigEarthNet the impact of data size on the relative performance of HMLC and MLC is not entirely clear; however, there seems to be a trend of HMLC being more helpful for small dataset sizes.

For the UCM dataset (Table 2), we overall can conclude that HMLC performs better than MLC across the different architectures, metrics, and especially in cases of limited data availability ($N \leq 500$). While ConvNeXt V2 generally achieves lower ranking loss and higher $\overline{\text{AUPRC}}$ values compared to DINOv2, incorporating hierarchical information brings more benefit to DINOv2, particularly with smaller dataset sizes, yielding consistent improvements of HMLC over MLC (with comparable performance at $N = 1000$).

### 5.2 Transfer Learning Experiments

To investigate whether HMLC improves over MLC in a transfer learning context, we conducted experiments using models that underwent intermediate fine-tuning

Table 2: Comparison between the predictive performance of different network architectures trained in MLC and HMLC settings on the UCM dataset. For each subset of data of size $N$, the best performance is highlighted in bold.

| $N$ | Task | $w_0$ | ConvNeXt V2 | | DINOv2 | |
|---|---|---|---|---|---|---|
| | | | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ |
| 100 | HMLC | 0.5 | **0.054** | 0.748 | 0.206 | 0.246 |
| | | 0.75 | 0.055 | **0.754** | **0.205** | **0.258** |
| | | 1.0 | 0.058 | 0.737 | 0.214 | 0.252 |
| | MLC | - | 0.061 | 0.752 | 0.206 | 0.243 |
| 250 | HMLC | 0.5 | **0.025** | **0.914** | 0.120 | 0.556 |
| | | 0.75 | 0.027 | 0.902 | **0.093** | **0.662** |
| | | 1.0 | 0.027 | 0.906 | 0.110 | 0.565 |
| | MLC | - | 0.031 | 0.913 | 0.173 | 0.359 |
| 500 | HMLC | 0.5 | 0.018 | 0.947 | **0.047** | 0.802 |
| | | 0.75 | **0.016** | **0.958** | **0.047** | **0.818** |
| | | 1.0 | 0.018 | 0.947 | 0.051 | 0.806 |
| | MLC | - | **0.016** | 0.957 | 0.080 | 0.653 |
| 1000 | HMLC | 0.5 | 0.013 | 0.962 | 0.051 | 0.789 |
| | | 0.75 | **0.012** | **0.967** | 0.048 | 0.787 |
| | | 1.0 | 0.014 | 0.966 | **0.037** | 0.827 |
| | MLC | - | **0.012** | 0.966 | 0.039 | **0.832** |

on the full set of images from the BigEarthNet dataset. Subsequently, the models were fine-tuned on the smaller, limited in size, UCM dataset for additional

Table 3: Comparison between the predictive performance of different network architectures in the transfer learning setting using the BigEarthNet-43 and BigEarthNet-19 models (trained on the whole dataset) and fine-tuned on the UCM dataset for both MLC and HMLC tasks. For each subset of data of size $N$, the best performance is highlighted in bold.

| $N$ | Task | $w_0$ | BigEarthNet-43 | | | | BigEarthNet-19 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | ConvNeXt V2 | | DINOv2 | | ConvNeXt V2 | | DINOv2 | |
| | | | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ | $rl(\downarrow)$ | $\overline{\text{AUPRC}}(\uparrow)$ |
| 100 | HMLC | 0.5 | 0.129 | 0.539 | 0.054 | 0.802 | **0.096** | **0.617** | 0.067 | 0.772 |
| | | 0.75 | 0.125 | 0.496 | 0.052 | 0.817 | 0.125 | 0.530 | **0.057** | **0.796** |
| | | 1.0 | 0.161 | 0.423 | 0.058 | 0.796 | 0.142 | 0.468 | 0.066 | 0.794 |
| | MLC | - | **0.097** | **0.626** | **0.049** | **0.831** | 0.126 | 0.514 | 0.060 | 0.787 |
| 250 | HMLC | 0.5 | **0.038** | 0.805 | 0.033 | 0.903 | 0.044 | **0.832** | 0.041 | 0.875 |
| | | 0.75 | 0.042 | 0.790 | **0.030** | **0.908** | 0.044 | 0.795 | **0.030** | **0.920** |
| | | 1.0 | 0.049 | 0.782 | 0.033 | 0.877 | 0.046 | 0.807 | 0.045 | 0.873 |
| | MLC | - | 0.039 | **0.812** | 0.032 | 0.898 | **0.042** | 0.803 | 0.031 | 0.896 |
| 500 | HMLC | 0.5 | **0.017** | 0.912 | 0.022 | 0.953 | **0.022** | **0.919** | 0.022 | 0.937 |
| | | 0.75 | 0.018 | 0.919 | **0.019** | **0.956** | 0.027 | 0.893 | **0.019** | 0.933 |
| | | 1.0 | 0.020 | 0.904 | 0.022 | 0.942 | 0.024 | 0.807 | 0.021 | 0.931 |
| | MLC | - | 0.018 | **0.922** | 0.020 | 0.944 | 0.027 | 0.908 | 0.021 | **0.941** |
| 1,000 | HMLC | 0.5 | **0.018** | 0.923 | 0.014 | 0.968 | **0.018** | **0.944** | 0.016 | 0.963 |
| | | 0.75 | 0.016 | 0.915 | 0.016 | 0.962 | 0.021 | 0.922 | **0.014** | **0.964** |
| | | 1.0 | 0.021 | 0.918 | **0.012** | 0.956 | 0.019 | 0.926 | 0.015 | 0.931 |
| | MLC | - | 0.021 | **0.946** | 0.013 | **0.970** | **0.018** | **0.944** | 0.016 | 0.959 |

20 epochs of training using 100, 250, 500 and 1000 of images. The results are summarized in Table 2.

The findings indicate that HMLC better adapts to the specific characteristics of the target domain compared to MLC. This holds primarily for the DINOv2 models from both hierarchies and for models pre-trained with BigEarthNet-19, with few exceptions where MLC outperforms HMLC. Conversely, ConvNeXt V2 achieves marginally better results in terms of ranking loss when using the BigEarthNet-43 hierarchy.

## 6 Conclusions

The topic of hierarchical multi-label classification has been poorly investigated in the remote sensing image classification domain due to the lack of appropriate datasets. In this work, we present hierarchical multi-label versions of the BigEarthNet and UCM datasets. These datasets can be a valuable resource for the remote sensing community in future method development and benchmarking endeavours. Furthermore, we show that by exploiting the hierarchical organization of classes, the classification methods can often achieve better predictive performance than the usual multi-label classification methods. More detailed experiments, past our initial investigation, are, however, needed to precisely investigate under which conditions and in which contexts hierarchical multi-label classification is the most appropriate.

**Acknowledgements** We acknowledge the support of the European Commission via the projects ASSAS (grant number 101059682), ELIAS (101120237), INQUIRE (101057499), PARC (101057014), and TAILOR (952215), as well as by the Slovenian Research Agency via the research program P2-0103 and the research projects J1-3033, J2-2505, J2-4452, J2-4660, J3-3070, J4-3095, J5-4575, J7-4636, J7-4637, and N2-0236.

## References

1. Sumbul, G., Charfuelan, M., Demir, B., Markl, V.: BigEarthNet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding. IEEE International Geoscience and Remote Sensing Symposium 12(2), 5901–5904 (2019)
2. Mehmood, M., Shahzad, A., Zafar, B., Shabbir, A., Ali, N.: Remote sensing image classification: A comprehensive review and applications. Mathematical Problems in Engineering 2022(1), 5880959 (2022)
3. Shao, C., Li, H., Shen, H.: Mctn-net: A multiclass transportation network extraction method combining orientation and semantic features. IEEE Geoscience and Remote Sensing Letters 21, 1–5 (2024)
4. Sumbul, G., Demİr, B.: A deep multi-attention driven approach for multi-label remote sensing image classification. IEEE Access 8, 95934–95946 (2020)
5. Stoimchev, M., Kocev, D., Džeroski, S.: Deep network architectures as feature extractors for multi-label classification of remote sensing images. Remote Sensing 15(2) (2023), https://www.mdpi.com/2072-4292/15/2/538

6. Shou Feng, P.F., Zheng, W.: A hierarchical multi-label classification method based on neural networks for gene function prediction. Biotechnology & Biotechnological Equipment 32(6), 1613–1621 (2018)
7. Gong, J., Teng, Z., Teng, Q., Zhang, H., Du, L., Chen, S., Bhuiyan, M.Z.A., Li, J., Liu, M., Ma, H.: Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. IEEE Access 8, 30885–30896 (2020)
8. Chen, H., Miao, S., Xu, D., Hager, G.D., Harrison, A.P.: Deep hiearchical multi-label classification applied to chest x-ray abnormality taxonomies. Medical Image Analysis 66, 101811 (2020)
9. Levatić, J., Kocev, D., Debeljak, M., Džeroski, S.: Community structure models are improved by exploiting taxonomic rank with predictive clustering trees. Ecological modelling 306, 294–304 (2015)
10. Levatić, J., Kocev, D., Džeroski, S.: The importance of the label hierarchy in hierarchical multi-label classification. Journal of Intelligent Information Systems 45(2), 247–271 (October 2015)
11. Noor, K.T., Robles-Kelly, A., Kusy, B.: A capsule network for hierarchical multi-label image classification. In: Krzyzak, A., Suen, C.Y., Torsello, A., Nobile, N. (eds.) Structural, Syntactic, and Statistical Pattern Recognition. pp. 163–172. Springer International Publishing, Cham (2022)
12. Zheng, Y., Chen, Q., Fan, J., Gao, X.: Hierarchical convolutional neural network via hierarchical cluster validity based visual tree learning. Neurocomputing 409, 408–419 (2020)
13. Kolisnik, B., Hogan, I., Zulkernine, F.: Condition-CNN: A hierarchical multi-label fashion image classification model. Expert Systems with Applications 182, 115195 (2021)
14. Zhang, S., Xu, R., Xiong, C., Ramaiah, C.: Use all the labels: A hierarchical multi-label contrastive learning framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16660–16669 (June 2022)
15. Chen, J., Qian, Y.: Hierarchical multilabel ship classification in remote sensing images using label relation graphs. IEEE Transactions on Geoscience and Remote Sensing 60, 1–13 (2022)
16. Taoufiq, S., Nagy, B., Benedek, C.: Hierarchynet: Hierarchical CNN-based urban building classification. Remote Sensing 12(22), 3794 (2020)
17. Sumbul, G., de Wall, A., Kreuziger, T., Marcelino, F., Costa, H., Benevides, P., Caetano, M., Demir, B., Markl, V.: BigEarthNet-MM: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. IEEE Geoscience and Remote Sensing Magazine 9(3), 174–180 (2021)
18. Chaudhuri, B., Demir, B., Chaudhuri, S., Bruzzone, L.: "Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method. IEEE Transactions on Geoscience and Remote Sensing 56(2), 1144–1158 (2018)
19. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: Convnext v2: Co-designing and scaling convnets with masked autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16133–16142 (2023)
20. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

## 6.2 Fully- and Semi-Supervised Hierarchical Multi-Label Image Classification with Graph Learning

The second study presents HELM, a unified framework that integrates hierarchical graph learning and self-supervised learning for HMLC. HELM models multiple paths within a hierarchy using graph-based reasoning while jointly optimizing supervised classification, graph consistency, and self-supervised objectives. The framework effectively leverages both labeled and unlabeled data, achieving strong results across eight datasets and outperforming state-of-the-art MLC and HMLC baselines.

The paper included in this section is:

- M. Stoimchev, B. Koloski, J. Levatić, D. Kocev, and S. Džeroski, "Fully- and semi-supervised hierarchical multi-label image classification with graph learning," under review for publication in *Pattern Recognition*, 2025.

**The contributions of Marjan Stoimchev to this paper are as follows.** Marjan Stoimchev designed and implemented HELM, including the multi-token transformer and graph-based hierarchical modules, integrated the self-supervised components, created the multi-path label structures where needed, conducted the supervised and semi-supervised experiments, carried out comparative and qualitative analyses, and drafted and revised the manuscript.

**Code availability.** All code developed for this study, including model implementations, training pipelines, evaluation scripts, and instructions for constructing the hierarchical datasets, is publicly available at: `https://github.com/marjanstoimchev/HELM`.

# Fully- and Semi-Supervised Hierarchical Multi-Label Image Classification with Graph Learning

Marjan Stoimchev[a,b,*], Boshko Koloski[a,b], Jurica Levatić[a], Dragi Kocev[a], Sašo Džeroski[a]

[a]*Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia*
[b]*Jožef Stefan International Postgraduate School, Ljubljana, Slovenia*

**Abstract**

Hierarchical multi-label classification (HMLC) is indispensable to model complex and structured label dependencies encountered in many domains of predictive modeling involving image analysis, such as remote sensing and medical diagnostics. Despite recent advances in deep learning and computer vision, existing methods face key limitations: they struggle to handle scenarios where instances belong to multiple paths in the hierarchy, fail to fully utilize hierarchical information, and largely focus on supervised learning tasks underutilizing unlabeled data. This paper introduces HELM, a novel hierarchical multi-label learning framework that overcomes these limitations. HELM introduces: *(i)* hierarchy-specific class tokens to capture internal label interactions within complex multi-path hierarchies; *(ii)* graph convolutional networks to encode hierarchical structures and generate hierarchy-aware embeddings; and *(iii)* a semi-supervised framework that leverages unlabeled data to enhance the discriminative power when labeled data is scarce. To facilitate advances in HMLC research, where image datasets are in short supply, we also construct HMLC versions for eight existing multi-label classification datasets from the remote sensing and medical domains. In a comprehensive evaluation on these datasets, HELM outperforms multi-label classification baselines by an average margin of 9.3% AU$\overline{\text{PRC}}$, achieves state-of-the-art performance as compared to existing HMLC methods, and improves upon its supervised baseline by exploiting unlabeled data. These results underscore HELM's robustness, adaptability to complex hierarchies, and effectiveness in leveraging unlabeled data to enhance learning.

*Keywords:* hierarchical multi-label classification, graph learning, semi-supervised learning, image classification

## 1. Introduction

Many real-world domains where predictive modeling is applied involve structured outputs where classes are interconnected [1]. These connections can be structured in the form of a tree-like hierarchy, a directed acyclic graph, or a sequence. Hierarchical multi-label classification (HMLC) deals specifically with problems where each sample is annotated with multiple labels, where all possible labels are arranged hierarchically. This setup is encountered in a range of applications, such as gene function prediction [1], text categorization [2], medical image analysis [3], taxonomy classification [4], and others. Previous research suggests that incorporating hierarchical information through HMLC can improve both predictive performance and efficiency as compared to multi-class classification (MCC), multi-label classification (MLC), and hierarchical single-label classification (HSLC) [5].

*Corresponding author.
*Email addresses:* marjan.stoimchev@ijs.si (Marjan Stoimchev), boshko.koloski@ijs.si (Boshko Koloski), jurica.levatic@ijs.si (Jurica Levatić), dragi.kocev@ijs.si (Dragi Kocev), saso.dzeroski@ijs.si (Sašo Džeroski)

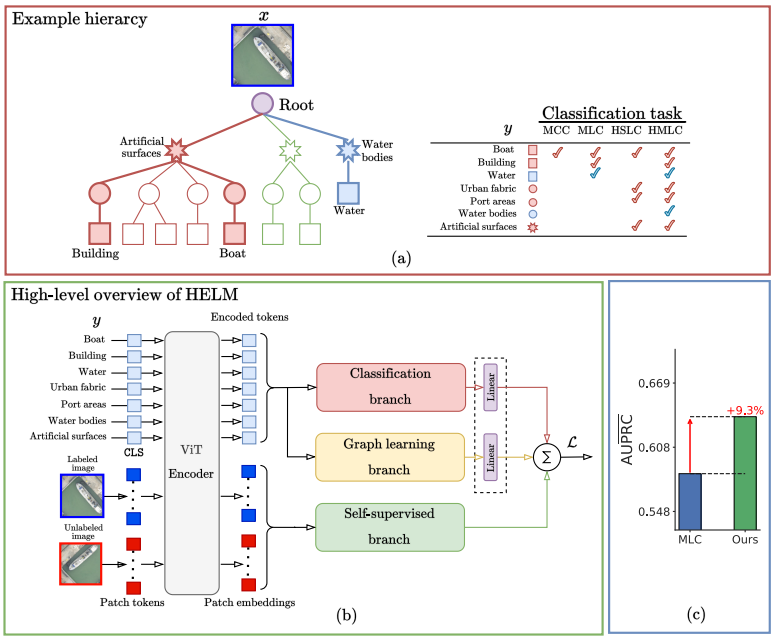



Figure 1: (a) Comparison of different image classification tasks. (b) A high-level overview of the proposed Hierarchical and Explicit Label Modeling (HELM) approach. (c) Average performance boost of HELM over the MLC baseline on eight benchmark datasets.

Figure 1(a) illustrates the differences between these predictive tasks. Briefly, for each sample, MCC predicts a single label from a set of (more than two) labels, MLC predicts multiple labels from such a set but disregards hierarchical label constraints, and HSLC predicts a single path from the root to a leaf label in the label hierarchy.

In the field of computer vision (CV), however, the advantages of exploiting hierarchical dependencies among labels within the HMLC framework remain largely unexplored. Researchers have mainly addressed the label hierarchy-agnostic tasks of MCC and MLC, while studies that account for label hierarchies are typically limited to HSLC, which predicts only a single path in the hierarchy [6].

While effective in certain scenarios, the HSLC-based methods do not fully exploit the hierarchical structure: They assume only one leaf label (at the lowest level of the hierarchy), whereas, as previously discussed, many relevant real-world settings, such as the analysis of medical or remotely sensed images, inherently involve multiple labels at the leaf level. A single method that tackles the HMLC task by considering clinical taxonomies of X-ray images has been tested on a single medical dataset, without exploration across different domains or datasets. The research on HMLC in the computer vision field is limited not only by the lack of studies addressing this task, but also by the shortage of image datasets annotated with labels organized in a hierarchy (with each dataset having its respective hierarchy).

Orthogonal to handling hierarchically organized classes, an additional challenge in predictive modeling is the limited availability of labeled data due to expensive and/or time-consuming annotation procedures. Consequently, labeled datasets are often small, making it difficult to build predictive models with satisfactory predictive performance. A common way to address this problem is semi-supervised learning (SSL), which leverages abundant unlabeled data to improve model performance [7].

The issue of tedious annotation of examples with class labels is even more pronounced for the HMLC task, as labels are structured and more complex than in usual predictive modeling scenarios. Consider the illustrative

example presented in Figure 1(a): In the classical multi-class scenario, an expert would have to verify only if a boat is present in the image, while in the HMLC scenario, seven different classes and sub-classes have to be identified. Thus, the problem of limited labeled data is amplified in the HMLC domain, highlighting the need for semi-supervised methods. Despite this, semi-supervised methods able to deal with the HMLC task are, to the best of our knowledge, practically non-existent in the computer vision field. The existing SSL methods are limited to MCC [8, 9] and MLC tasks [10].

Following the identified inherent challenges of predictive modeling for images, there is a pressing need for computer vision methods capable of handling samples associated with multiple hierarchically organized labels, while effectively leveraging unlabeled data. To respond to these needs of the field, we propose the *Hierarchical and Explicit Label Modeling* (HELM) method – a semi-supervised learning framework designed for hierarchical multi-label classification. HELM utilizes a multi-branch architecture that explicitly incorporates the known label hierarchy to model relationships between labels and enhances representation learning by integrating both labeled and unlabeled data.

As illustrated in Figure 1(b), HELM consists of three key components: *(i)* a classification branch that leverages labeled data to enhance the discriminative power of the learned embeddings; *(ii)* a graph learning branch that employs a graph convolutional network (GCN) to explicitly capture hierarchical dependencies, ensuring structure-aware representations while preserving label relationships; and *(iii)* a self-supervised branch that exploits unlabeled data to refine feature extraction and improve generalization. By jointly optimizing these components, HELM ensures hierarchy-consistent predictions and maximizes information utilization, making it highly effective for complex HMLC tasks.

Through detailed experimental evaluation, both qualitative and quantitative, we show that the proposed method achieves superior predictive performance by successfully leveraging the label hierarchy and unlabeled examples in datasets from various domains. Moreover, HELM achieves consistent improvements over the current state-of-the-art approaches.

The main contributions of our work are summarized as follows:

- We introduce HELM, a method that models multiple paths within an explicit label hierarchy and seamlessly incorporates unlabeled data to improve predictive performance. To the best of our knowledge, this is the first semi-supervised image-based method capable of modeling multi-path labels in class hierarchies.
- We propose a novel multi-token transformer architecture that integrates hierarchical dependencies through graph-based learning while incorporating hierarchy-specific class tokens into the Vision Transformer (ViT) encoder to capture internal label interactions. By jointly optimizing classification, graph learning, and self-supervised learning, HELM effectively balances predictive performance and hierarchy consistency.
- Through extensive experiments on eight real-world datasets, we demonstrate that HELM consistently outperforms its MLC baseline, highlighting the benefits of utilizing hierarchical information during training. Furthermore, we show that HELM is superior in predictive performance compared to state-of-the-art methods from the literature and outperforms its supervised baseline in semi-supervised learning scenarios.
- We provide eight new HMLC image benchmark datasets by constructing label hierarchies and providing multi-path image annotations for existing MLC datasets from remote sensing and medical domains.

## 2. Related Works

### *2.1. Hierarchical image classification*

HMLC focuses on problems where each instance can be assigned multiple hierarchically arranged labels. In real-world applications, this hierarchical structure can be a tree or a directed acyclic graph. HMLC arises naturally in text categorization [2], functional genomics [1], and image classification tasks, including medical image annotation [3] and taxonomic image classification [11].

Modern methods based on deep neural networks (DNNs) typically adopt one of two main strategies to deal with the HMLC task: *(i)* translating the hierarchy into the network design [12, 13, 14], or *(ii)* embedding hierarchical constraints into the loss function [6, 15].

Methods that align the network structure with the class hierarchy include the Branch Convolutional Neural Networks (B-CNNs) [13], which extend traditional CNNs with multiple output layers. This enables predictions that progress from coarse to fine-grained categories while leveraging prior hierarchical knowledge to improve classification performance. However, one of the major difficulties in B-CNN's design is the computational complexity introduced by the additional branches used to model the hierarchical relationships. To address limitations in training time and accuracy, Condition-CNN [12] was introduced, using the Teacher Forcing training algorithm to condition each lower-level branch on the correct predictions from higher levels. This design prevents error propagation and reduces computational overhead by considering inter-level class relationships as conditional probabilities. However, approaches in this category may not scale well when the hierarchy becomes deeper or more complex.

The other line of research embeds hierarchical knowledge directly into the loss function to enforce label dependencies without rigidly coupling them to the network design. Chen et al. [16] proposed a Hierarchical Residual Network (HRN) that treats parent-level features as residual connections for child-level predictions. Their loss combines information from related hierarchical labels, adding a multi-class cross-entropy term to improve leaf-level predictions. Another example is the use of hierarchy and exclusion (HEX) graphs [17], where a two-channel output structure captures both the hierarchical relationships among labels and the fine-grained distinctions at the lowest level. Zhang et al. [6] extended this idea through a two-stage framework called HiMulConE. They first learn label-aware embeddings with a hierarchy-preserving contrastive loss and then train a leaf-level classifier using cross-entropy, but still assume a single path in the class hierarchy.

While most DNN methods that use the label hierarchy are limited to a single hierarchical path per sample, Coherent HMC Neural Network (C-HMCNN) [15] allows for multiple paths and thus handles more complex annotations. C-HMCNN modifies the binary cross-entropy (BCE) loss to ensure parent-child consistency and reuses lower-level predictions to improve higher-level classification. This strategy accommodates more flexible label assignments, making it suitable for cases where samples can belong to multiple leaf-level categories. C-HMCNN was, however, developed for tabular data.

Recent work has shown that Graph Neural Networks (GNNs) can naturally model label dependencies: Class hierarchies and relationships can be represented as graphs, enabling GNNs to capture long-range dependencies more effectively. For example, Chen et al. [18] used a Graph Convolutional Network (GCN) to learn inter-dependent label classifiers, where each node in a directed graph of object labels is represented by word embeddings. Lan et al. [19] proposed a GCN-based model for multi-label attribute classification in anime illustrations.

By leveraging WordNet [20] to build attribute hierarchies, they introduced semantic features derived from both real-world and animated-style images, as well as a hierarchical loss that enforces attribute relationships to improve overall classification accuracy.

To overcome the limitations of existing HMLC methods, we introduce a framework that explicitly integrates hierarchical dependencies through Graph Convolutional Networks (GCNs) while leveraging hierarchy-specific class tokens to encode label interactions. Unlike prior approaches that enforce hierarchical constraints through loss functions or rigidly translate the hierarchy into the network design, our method learns hierarchy-aware embeddings by propagating label relationships within a structured graph. Importantly, it moves beyond the common single-path assumption, naturally accommodating multi-path annotations, where instances belong to multiple branches. By unifying transformer-based feature representation with graph-driven hierarchical reasoning, our approach better reflects real-world classification challenges, ensuring predictions that are coherent, interpretable, and structurally consistent.

### *2.2. Semi-supervised learning*

In the field of semi-supervised learning, various strategies have been proposed to leverage unlabeled data effectively while minimizing the reliance on large labeled datasets. One representative approach is the Pseudo-Label method [9], which iteratively assigns labels to unlabeled data based on the model's high-confidence predictions. A closely related strategy is called Noisy Student [21], where a "teacher" model generates pseudo-labels for unlabeled samples, and a "student" model is trained on these labels with added stochastic noise for improved robustness. Another important approach is co-training, originally introduced by Blum et al. [22], where multiple models are trained in parallel and only those unlabeled samples on which all models agree are retained for further training. Beyond these label-based techniques, consistency regularization focuses on enforcing stable predictions under various perturbations; a notable example is FixMatch [8], which combines a pseudo-labeling strategy with strong augmentation to refine the model's predictions.

While the above SSL methods have proven effective for the MCC task, they often lack adaptability to more complex prediction tasks, such as HMLC, where label dependencies require specialized modeling approaches. Moreover, available implementations of methods for these more complex tasks remain limited or nonexistent, highlighting a gap in the practical adoption of SSL techniques beyond the standard MCC settings. For a comprehensive overview of existing SSL methods in the broader landscape, the reader is referred to a survey article on SSL [7].

Beyond the semi-supervised framework, self-supervised learning has emerged as another powerful paradigm for leveraging large amounts of unlabeled data. By introducing "pretext tasks" that encourage the model to learn useful representations, self-supervised learning significantly reduces the reliance on labeled data for subsequent fine-tuning. One set of self-supervised methods focuses on context-based tasks, such as rotation prediction (where the network learns to identify the correct orientation of images), colorization (reconstructing color information from grayscale input), and Jigsaw puzzles (reassembling shuffled image patches). These tasks help the model learn spatial, contextual, and semantic patterns within the image data. In contrast, contrastive learning methods prompt the model to produce invariant representations by comparing different augmented views of the image. Meanwhile, generative approaches, such as the masked autoencoder (MAE) approach [23], reconstruct masked parts of the input to capture fine-grained details.

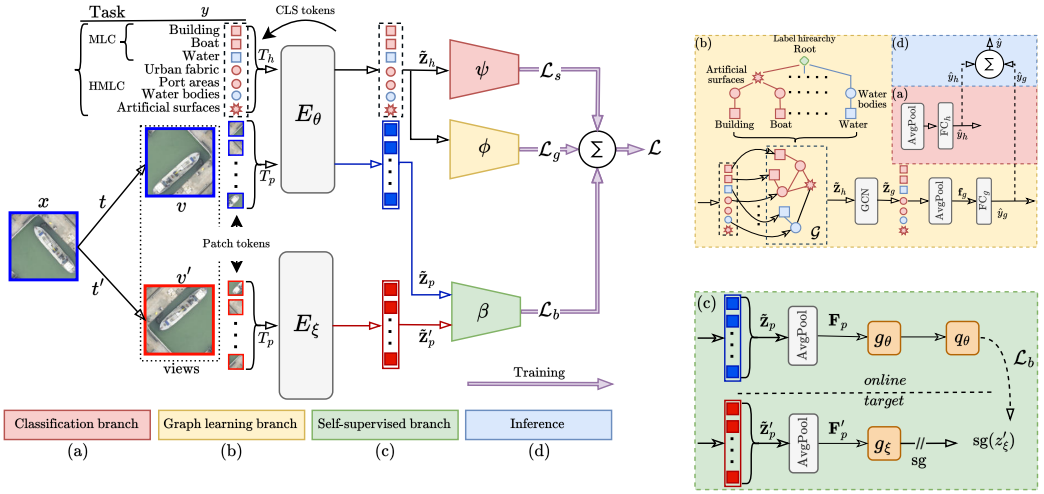



Figure 2: Overview of the HELM method. The framework consists of a transformer encoder with hierarchy-specific tokens that interact with patch tokens to produce label-specific embeddings, used across three branches: (a) classification branch, which addresses the hierarchical multi-label classification (HMLC) task by flattening the hierarchy into a multi-label classification (MLC) problem; (b) graph learning branch, which propagates label dependencies using a Graph Convolutional Network (GCN); and (c) the self-supervised branch, which employs BYOL for learning feature representations from unlabeled data. The model integrates these components through a joint learning process by summing the losses from each branch to optimize the parameters.

Self-supervised learning has gained massive popularity in fields such as remote sensing [24] and medical image classification [25], where collecting labeled data can be both expensive and time-consuming. The scarcity of expert annotations in these domains often creates bottlenecks that make purely supervised solutions impractical. By allowing models to learn from unlabeled data at scale, self-supervised approaches reduce the need for large labeled datasets and therefore align closely with the goals of SSL. According to Van Engelen et al. [7], self-supervised learning can be seen as an unsupervised preprocessing step that produces feature representations for later fine-tuning. Following this two-stage strategy frequently yields robust embeddings that transfer well across tasks, including MCC and MLC [24]. Despite the growing interest in semi-supervised learning, HMLC remains largely unexplored, with no existing approaches specifically addressing the challenges in computer vision.

Unlike the previously discussed methods, HELM is an end-to-end framework explicitly tailored for hierarchical multi-label image classification in both supervised and semi-supervised settings. To capture internal label dependencies, HELM introduces hierarchy-specific class tokens, while a GCN-based structure encoder learns hierarchy-aware embeddings that preserve label relationships. Furthermore, BYOL [26] is integrated as a self-supervised component, allowing HELM to use unlabeled data alongside limited labeled samples. This synergy enhances the discriminative power of the learned representations and makes HELM well-suited for SSL tasks involving complex label hierarchies.

## 3. Methodology

The architecture of the proposed transformer-based framework, illustrated in Figure 2, consists of three key components: *(i)* a discriminative learning branch dedicated to addressing the hierarchical multi-label classification (HMLC) task (Figure 2a), *(ii)* a graph learning branch, which leverages explicit hierarchical information to guide the generation of hierarchy-aware label embeddings, and *(iii)* a self-supervised learning branch, designed to leverage the information from unlabeled data, thereby guiding the model in generating improved representations

for the classification task.

In the subsequent subsections, we describe in detail how these components contribute to effective HMLC.

### *3.1. Encoder*

The encoder serves as the foundational component of our HELM framework, transforming input images into rich hierarchical representations that capture both visual semantics and label structure dependencies. Our approach extends the ViT architecture with hierarchy-specific class tokens to enable explicit modeling of hierarchical label relationships.

Given an input image $x \in \mathbb{R}^{C\times W\times H}$, we follow the standard ViT tokenization process by dividing it into $N_p$ non-overlapping patches of size $P \times P$, where $N_p = HW/P^2$. Each patch is linearly projected to obtain a sequence of patch tokens $T_p \in \mathbb{R}^{N_p\times d}$, where $d$ is the token embedding dimension, $C$ is the number of image channels, and $W$ and $H$ represent the image width and height, respectively.

To incorporate hierarchical label constraints directly into the encoder architecture, we introduce $M$ hierarchy-specific CLS tokens $T_{\texttt{CLS}} \in \mathbb{R}^{M\times d}$, where $M = M_l + M_h$ denotes the total number of tokens corresponding to both leaf labels ($M_l$) and intermediate hierarchy nodes ($M_h$). While the specific construction of the label hierarchy will be detailed in the experimental section, these specialized tokens are strategically designed to serve dual purposes within our framework: *(i)* they provide the appropriate output dimensionality for the classification branch ($M_l$ outputs for multi-label classification (MLC) tasks focusing on leaf labels, or $M$ outputs for HMLC tasks encompassing all hierarchy nodes), and *(ii)* they serve as learnable node embeddings for the graph learning branch, where each token corresponds to a specific node in the label hierarchy graph. This design enables the encoder to learn representations that capture both label-specific visual features and inter-label dependencies, providing a unified foundation for both supervised classification and graph-based hierarchical reasoning.

The hierarchy-specific CLS tokens are prepended to the patch tokens to form the complete input sequence:

$$T = [T_{\texttt{CLS}} \,\|\, T_p] \in \mathbb{R}^{(M+N_p)\times d} \tag{1}$$

This combined sequence is processed through a ViT encoder $E(\cdot, \theta)$ to produce contextualized representations:

$$\tilde{\mathbf{z}} = E(T, \theta) \in \mathbb{R}^{(M+N_p)\times d}, \tag{2}$$

where $\theta$ denotes the trainable parameters of the encoder. The hierarchy-specific CLS token outputs from $\tilde{\mathbf{z}}$ are subsequently utilized by both the classification and graph learning branches, providing a shared foundation for supervised learning and hierarchical reasoning.

### *3.2. Classification Branch*

The classification branch serves as the primary supervised learning component of our framework, directly mapping visual features to hierarchical label predictions. This branch operates exclusively on labeled samples to provide strong supervised signals for learning discriminative representations.

Let $\tilde{\mathbf{z}}_{\texttt{CLS}} \in \mathbb{R}^{M\times d}$ be the subset of $\tilde{\mathbf{z}}$ (Eq. 2) corresponding to the $M$ hierarchy-specific CLS tokens. We aggregate these embeddings through average pooling to obtain a unified representation:

$$\mathbf{f}_{\texttt{CLS}} = \frac{1}{M}\sum_{m=1}^{M} \tilde{\mathbf{z}}_{\texttt{CLS}}^{(m)} \in \mathbb{R}^{d}. \tag{3}$$

The aggregated representation $\mathbf{f}_{\texttt{CLS}}$ is then projected through a fully connected layer $\text{FC}_{\texttt{CLS}}$ onto the target label space, producing logits $\hat{y}_{\texttt{CLS}}$ that are converted to probabilities via sigmoid activation. The output dimensionality depends on the task formulation: for HMLC, the branch predicts both leaf and intermediate labels, while for standard MLC, it focuses exclusively on leaf labels.

The classification loss operates solely on labeled samples with batch size $B_l$:

$$\mathcal{L}_s = \frac{1}{B_l} \sum_{i=1}^{B_l} \mathcal{H}\Big(y_i^{(t)}, p^{(t)}(y|x_i)\Big), \tag{4}$$

where $\mathcal{H}$ denotes binary cross-entropy, $t \in \{\text{MLC}, \text{HMLC}\}$ is the task, and $p^{(t)}(y|x_i)$ represents the predicted probabilities for sample $x_i$ under task $t$.

This design ensures that the classification branch provides reliable supervised learning signals while remaining flexible to different hierarchical modeling requirements. By processing only labeled data, it maintains prediction quality and serves as a stable foundation for the overall framework, complementing the graph learning and self-supervised branches that exploit the availability of additional data.

### *3.3. Graph Learning Branch*

To exploit hierarchical dependencies among labels and enable effective knowledge propagation, we construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from the label hierarchy, where $\mathcal{V}$ corresponds to all labels (leaf and intermediate), and $\mathcal{E}$ encodes their parent-child relationships. This graph-based approach serves a dual purpose: *(i)* modeling structural label dependencies through hierarchical relationships, and *(ii)* enabling semi-supervised learning by propagating supervision signals from labeled to unlabeled samples through the graph topology.

We use the hierarchy-specific CLS tokens $\tilde{\mathbf{z}}_{\texttt{CLS}}$ as initial node features and apply a GraphSAGE convolutional operator $\phi(\cdot; \mathcal{G})$ to capture both local neighborhood information and global hierarchical constraints:

$$\tilde{\mathbf{z}}_g = \phi\Big(\tilde{\mathbf{z}}_{\texttt{CLS}}; \mathcal{G}\Big) \in \mathbb{R}^{M \times d}. \tag{5}$$

The GraphSAGE operator performs neighborhood aggregation and message passing between connected nodes in the label hierarchy. This mechanism allows the model to leverage hierarchical constraints and propagate supervision signals across the label structure, which is particularly effective in semi-supervised scenarios where unlabeled samples benefit from the structural knowledge encoded in the hierarchy.

We then apply average pooling across the $M$ graph-processed tokens to produce hierarchy-aware representations:

$$\mathbf{f}_g = \frac{1}{M} \sum_{m=1}^{M} \tilde{\mathbf{z}}_g^{(m)} \in \mathbb{R}^d, \tag{6}$$

and project $\mathbf{f}_g$ via a fully connected layer $\text{FC}_g$ to produce logits $\hat{y}_g$ that are converted to probabilities using sigmoid activation.

The graph branch operates on the entire batch (both labeled and unlabeled samples) to maximize the benefit of hierarchical knowledge propagation. However, crucially, the graph loss is computed exclusively on labeled samples to maintain supervised learning integrity:

$$\mathcal{L}_g = \frac{1}{B_l} \sum_{i=1}^{B_l} \mathcal{H}\Big(y_i,\ p_g(y|x_i)\Big), \tag{7}$$

where $\mathcal{H}$ denotes binary cross-entropy, and the loss is normalized by the number of labeled samples $B_l$ rather than the total batch size $B = B_l + B_u$.

This design enables the graph branch to perform dual function: *(i)* in supervised setting, it enhances label prediction through hierarchical dependency modeling and constraint enforcement, and *(ii)* in semi-supervised setting, it leverages unlabeled samples for improved representation learning while maintaining supervised prediction quality. The unlabeled samples contribute to feature enhancement through graph-based knowledge propagation without introducing noise in the loss computation, effectively enabling semi-supervised graph learning to benefit from the structural knowledge of the label hierarchy.

### *3.4. Self-Supervised Branch*

To leverage both labeled and unlabeled data for improved representation learning, we integrate the BYOL *(Bootstrap Your Own Latent)* framework [26] as our self-supervised learning component. This branch operates on the entire training batch, treating all images (both labeled and unlabeled) in a label-agnostic manner to learn robust visual representations through contrastive learning.

The BYOL framework consists of two networks: an online network parameterized by $\theta$ (shared with the main encoder) and a target network parameterized by $\xi$. For each image in the batch, we generate two augmented views $\mathbf{v}$ and $\mathbf{v}'$ through different data augmentation strategies. These views are processed through their respective encoders to extract patch token representations, which are then average-pooled to obtain feature representations $\mathbf{F}_p$ (from the online network) and $\mathbf{F}'_p$ (from the target network).

The online network applies a projection head $g_\theta$ followed by a predictor network $q_\theta$ to its features, while the target network only applies its projection head $g_\xi$ without a predictor. The self-supervised loss encourages the online network to predict the target network's representation of the same image under different augmentations:

$$\mathcal{L}_b = 2 - 2 \cdot \frac{\langle q_\theta(g_\theta(\mathbf{F}_p)), g_\xi(\mathbf{F}'_p)\rangle}{\left\|q_\theta(g_\theta(\mathbf{F}_p))\right\| \cdot \left\|g_\xi(\mathbf{F}'_p)\right\|}. \tag{8}$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product and $\|\cdot\|$ the L2 norm. The target network parameters are updated through exponential moving averaging: $\xi \leftarrow \tau\xi + (1-\tau)\theta$, where $\tau \in [0, 1)$ is a momentum coefficient that ensures stable learning dynamics.

Crucially, this self-supervised objective operates independently of label information, enabling the model to learn meaningful visual representations from the entire dataset. This approach is particularly valuable in semi-supervised scenarios where unlabeled samples significantly outnumber labeled ones, as it allows the encoder to capture rich visual semantics that enhance both the classification and graph learning branches.

### *3.5. Overall Objective*

The final training loss combines the classification branch, graph branch, and self-supervised branch:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_g + \mathcal{L}_b, \tag{9}$$

where $\mathcal{L}_s$, $\mathcal{L}_g$, and $\mathcal{L}_b$ represent the classification, graph, and self-supervised losses, respectively. Each component can be selectively included based on the training configuration. By unifying hierarchical classification, graph-based dependency modeling, and self-supervised feature learning, the proposed framework effectively leverages both labeled and unlabeled data to achieve robust performance.

Table 1: Summary of the datasets used in this study. $N$ denotes the total number of images in each dataset, while $N_{train}$ and $N_{test}$ represents the number of images in the training and test sets, respectively. $|\mathcal{L}|$ indicates the number of unique labels at each hierarchical level, where $\ell$ corresponds to the deepest (leaf) level and $h$ refers to overall dataset-level characteristics. *Card*@Level represents label cardinality (the average number of labels per image) for each level, and *Dens*@Level denotes label density (the average proportion of images with a given label) at each level. These metrics are provided for each hierarchical level, offering insights into the label structure and distribution within each dataset.

| | $N$ | $N_{train}$ | $N_{test}$ | $\lvert\mathcal{L}\rvert$ | | | | | | | | *Card*@Level | | | | | | | | *Dens*@Level | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | | | | 1 | 2 | 3 | 4 | 5 | 6 | $\ell$ | $h$ | 1 | 2 | 3 | 4 | 5 | 6 | $\ell$ | $h$ | 1 | 2 | 3 | 4 | 5 | 6 | $\ell$ | $h$ |
| UCM | 2,100 | 1,667 | 433 | 4 | 9 | 17 | - | - | - | 17 | 30 | 1.740 | 3.031 | 3.189 | - | - | - | 3.189 | 7.960 | 0.435 | 0.337 | 0.188 | - | - | - | 0.188 | 0.265 |
| AID | 3,000 | 2,400 | 600 | 4 | 9 | 17 | - | - | - | 17 | 30 | 2.175 | 4.513 | 4.933 | - | - | - | 4.933 | 11.621 | 0.544 | 0.501 | 0.290 | - | - | - | 0.290 | 0.387 |
| DFC-15 | 3,341 | 2,672 | 669 | 3 | 7 | 7 | - | - | - | 8 | 17 | 1.607 | 2.726 | 2.497 | - | - | - | 2.795 | 6.830 | 0.536 | 0.389 | 0.357 | - | - | - | 0.349 | 0.402 |
| MLRSNet | 109,151 | 87,336 | 21,815 | 7 | 15 | 22 | 60 | - | - | 60 | 104 | 2.206 | 3.617 | 4.414 | 5.012 | - | - | 6.099 | 15.248 | 0.315 | 0.241 | 0.201 | 0.098 | - | - | 0.102 | 0.161 |
| MuRed | 2,208 | 1,764 | 444 | 4 | 8 | 17 | 5 | - | - | 20 | 34 | 1.131 | 1.211 | 1.043 | 0.217 | - | - | 1.267 | 3.602 | 0.283 | 0.151 | 0.061 | 0.043 | - | - | 0.063 | 0.106 |
| HPA | 31,072 | 24,837 | 6,235 | 6 | 28 | - | - | - | - | 28 | 34 | 1.565 | 1.632 | - | - | - | - | 1.632 | 3.197 | 0.261 | 0.058 | - | - | - | - | 0.058 | 0.094 |
| ChestX-ray14 | 112,120 | 86,524 | 25,596 | 4 | 4 | 13 | - | - | - | 20 | 21 | 1.028 | 1.109 | 0.627 | - | - | - | 1.221 | 2.175 | 0.206 | 0.158 | 0.039 | - | - | - | 0.061 | 0.081 |
| PadChest | 121,230 | 97,203 | 24,027 | 2 | 5 | 9 | 9 | 7 | 2 | 19 | 34 | 1.000 | 1.026 | 0.238 | 0.123 | 0.055 | 0.017 | 1.036 | 2.452 | 0.500 | 0.205 | 0.026 | 0.014 | 0.008 | 0.009 | 0.054 | 0.074 |

During inference (Figure 2d), the self-supervised branch is omitted. The logits from the classification and graph branches, $\hat{y}_s$ and $\hat{y}_g$, are combined differently based on the task formulation. For HMLC, both branches contribute: $\hat{y} = \sigma(\hat{y}_s + \hat{y}_g)$. For standard MLC or when graph learning is not applied, only the classification branch is used: $\hat{y} = \sigma(\hat{y}_s)$. The key distinction is that MLC focuses exclusively on leaf labels from the hierarchy, while HMLC considers all labels (leaf and intermediate) but treats them independently without enforcing hierarchical constraints during prediction.

## 4. Experimental Design

### *4.1. Datasets*

We evaluated the proposed method using four publicly available land use and land cover remote sensing image (RSI) datasets: UCM [27], AID [28], DFC-15 [29], and MLRSNet [30]; and four medical imaging datasets: MuRed [31], HPA [32], PadChest [33], and NIH ChestX-ray14 dataset [34]. These datasets are originally MLC, meaning that multiple labels are associated with each sample, but the hierarchical dependencies among classes were not explicitly defined (even though such dependencies inherently exist).

Since comprehensive HMLC image datasets are limited, we adapted these datasets to the hierarchical multi-label setting. For the RSI datasets, we constructed hierarchical label structures by systematically mapping the multi-label classes to the CORINE Land Cover (CLC) nomenclature [35]. The CLC provides a comprehensive and standardized taxonomy of land cover classes across multiple hierarchical levels, enabling us to create meaningful hierarchical relationships while preserving the multi-label nature at the leaf level. For the medical imaging datasets, we organized the multi-label medical conditions into hierarchical structures based on the International Classification of Diseases, 10th Revision (ICD-10) codes [36] – a clinically validated classification system. This approach ensures that the resulting hierarchies reflect genuine medical taxonomic relationships while maintaining the multi-label characteristics essential for realistic diagnostic scenarios.

When direct mapping to established nomenclatures was not feasible due to dataset-specific terminologies or emerging categories, we supplemented our approach by querying ChatGPT, followed by a manual inspection, to generate appropriate hierarchical placements. Figure 3 illustrates representative examples of the constructed hierarchies, demonstrating how the CORINE Land Cover nomenclature and ICD-10 codes translate into structured taxonomic relationships for the remote sensing and medical domain datasets, respectively. Table 1 presents the detailed characteristics of the constructed label hierarchies.

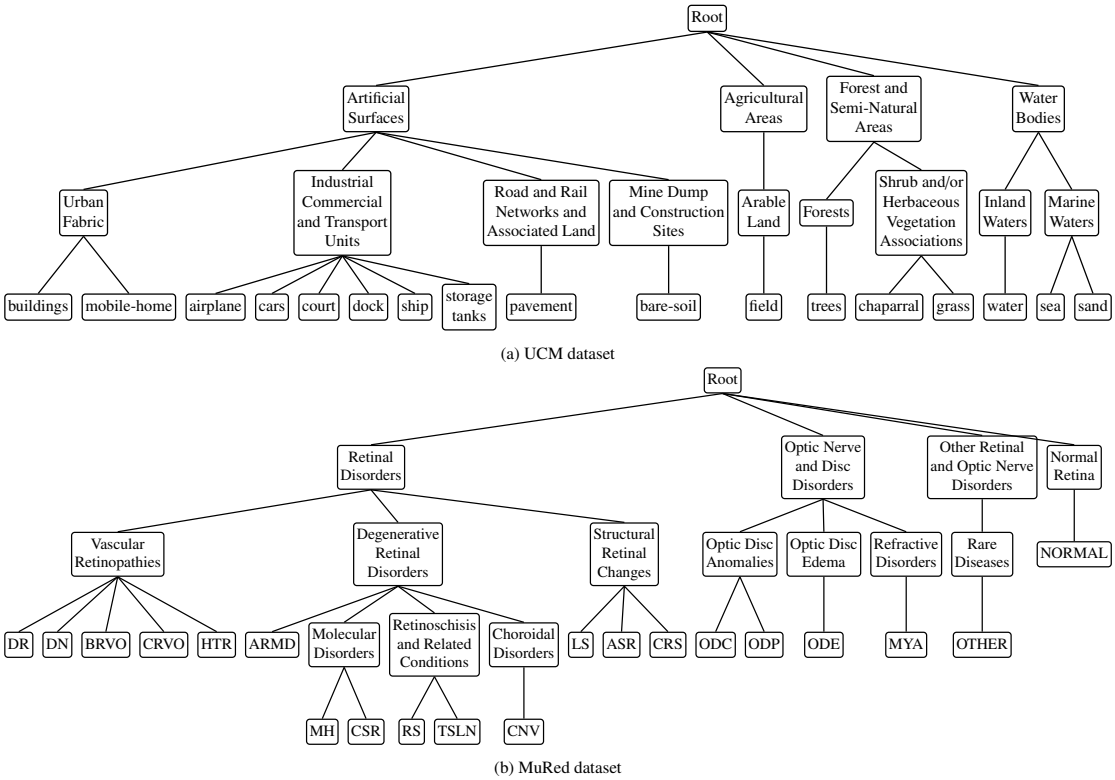



Figure 3: Examples of constructed label hierarchies for (a) the UCM dataset, derived from the CORINE Land Cover (CLC) nomenclature, and (b) the MuRed dataset, with abbreviations in leaf labels correspond to ICD-10 codes of disease names.

#### *4.1.1. Remote Sensing Image Datasets*

The MLC version of the UCM dataset comprises 2,100 aerial RGB images with a resolution of 256×256 pixels and 17 object-level labels [27]. The HMLC extension of UCM includes a total of 30 labels distributed across three hierarchical levels, providing a more detailed classification structure. AID consists of 3,000 aerial images with a resolution of $600 \times 600$ pixels, categorized into 17 object-level labels [28]. Similarly, the HMLC extension of AID contains 30 labels organized across three hierarchical levels. The DFC-15 dataset, originating from the 2015 IEEE GRSS Data Fusion Contest, is derived from a semantic segmentation dataset [29]. It comprises 3,342 image patches with a resolution of $600 \times 600$ pixels, grouped into eight distinct object categories. The HMLC version of this dataset introduces a three-level hierarchy with 17 distinct labels. Lastly, the MLRSNet dataset includes 109,161 images annotated into 60 categories, each with a resolution of $256 \times 256$ pixels [30]. The hierarchical version of MLRSNet expands this structure to include 104 labels organized across four hierarchical levels, making it the most complex and challenging dataset among those considered in this study.

#### *4.1.2. Medical Image Datasets*

The first medical imaging dataset is the Multi-label Retinal Disease (MuRed) dataset [31], a publicly available collection of 2,208 retinal fundus images for multi-label classification of retinal diseases. It addresses limitations of existing datasets by providing broader pathology coverage, improved class balance, and higher-quality samples. The dataset comprises 20 disease labels and a hierarchical extension with 34 labels structured across four levels.

The Human Protein Atlas (HPA) Image Classification dataset [32], originally developed for a Kaggle competition, focuses on predicting protein structures within cellular images. This dataset features 28 distinct labels and supports multi-label classification since individual images can contain multiple proteins simultaneously. The collection comprises 31,072 samples acquired through standardized confocal microscopy protocols. Each sample consists of four imaging channels: the target protein (green) alongside three cellular reference markers consisting of the nucleus (blue), microtubules (red), and endoplasmic reticulum (yellow). Original image dimensions range from $2048 \times 2048$ to $3072 \times 3072$ pixels. In our experimental setup, we utilize only the RGB channels and standardize all images to $224 \times 224$ pixel resolution. For hierarchical multi-label classification experiments, we extend the original dataset by organizing the labels into a two-level hierarchy containing 34 total labels.

The PadChest dataset is a large-scale chest X-ray dataset comprising 160,868 images from 67,000 patients, acquired at Hospital San Juan (Spain) between 2009 and 2017 [33]. It includes six different position views and metadata on image acquisition and patient demographics. The original label space contains 193 labels, including 174 radiographic findings, 19 differential diagnoses, and 104 anatomical locations, organized hierarchically and mapped to the Unified Medical Language System (UMLS). Approximately 27% of the labels were manually annotated by board-certified radiologists, with the remaining labels generated using a recurrent neural network with attention mechanisms. We utilized the predefined taxonomy from [37], which organizes 121,242 samples into a hierarchy of 32 labels across six hierarchical levels.

The National Institutes of Health (NIH) ChestX-ray14 dataset is an extension of the ChestX-ray8 dataset [34], expanding the label set to include 14 pathologies and an additional "No Finding" label, resulting in a total of 15 labels. Following [38], we explicitly encode the "No Finding" label as an additional category. The dataset comprises 112,120 frontal-view chest X-ray images from 32,717 unique patients, each with a resolution of $1024 \times 1024$ pixels. Additional patient metadata, such as gender and age, is also provided, making it a valuable resource for MLC in medical imaging. We further extend this dataset by incorporating hierarchical information, resulting in a taxonomy of 22 labels across three hierarchical levels.

### *4.2. Implementation Details*

Our HELM framework is implemented using the PyTorch framework, leveraging the base variant of the Vision Transformer (ViT-B/16) with a patch size of $16 \times 16$ pixels. The ViT backbone is initialized with pre-trained ImageNet weights to benefit from rich visual representations. For the graph learning component, we employ PyTorch Geometric (PyG)[1] with a two-layer GraphSAGE convolutional network, enabling efficient neighborhood aggregation and message passing within the label hierarchy. The self-supervised BYOL component is integrated using LightlySSL[2], a specialized framework for self-supervised learning in computer vision.

Models are trained on RGB images resized to $224 \times 224$ pixels using the unified learning objective defined in Eq. 9. The hierarchy-specific CLS tokens produce latent representations of dimension $d = 768$ after average pooling. For the supervised components (classification and graph branches), we apply weak augmentations, including horizontal and vertical flips and Gaussian blur. For the BYOL component, we employ both weak and strong augmentations, where strong augmentations additionally include color jittering, affine transformations, random

[1] `https://pytorch-geometric.readthedocs.io/en/latest/`
[2] `https://github.com/lightly-ai/lightly`

cropping, and random erasing to enhance self-supervised representation learning.

We trained the models for 100 epochs on four NVIDIA A100 GPUs equipped with 40 GB memory each, employing the AdamW optimizer. We used a base learning rate of $1 \times 10^{-4}$ with a cosine annealing scheduler and a mini-batch size of 16. Mixed-precision training is utilized to accelerate computations and reduce memory consumption.

### *4.3. Evaluation Strategy*

To assess the effectiveness of our approach, we utilize a separate fixed-size test set to measure the performance of the final model. We use predefined test sets where available; otherwise, we partition the datasets into training and testing subsets using an iterative stratified sampling strategy to preserve the relative frequency of labels. The resulting splits allocate approximately 80% of the data to training and 20% to testing, with a validation set comprising approximately 10% of the training data. For the PadChest and ChestX-ray14 datasets, we adhere to the official splits to ensure no patient overlap between training, validation, and test sets.

To evaluate the semi-supervised learning capabilities of our framework, we employ an inductive learning setting where models are trained to predict labels for previously unseen examples. We systematically control the proportion of labeled supervision by randomly selecting subsets of the training set in the percentages $\{1, 5, 10, 25\}$ as labeled samples, while treating the remaining data as unlabeled by removing their ground-truth annotations. This experimental design allows us to assess how effectively our method can leverage abundant unlabeled data when only limited labeled supervision is available. For each labeling proportion, we perform three independent experiments with different random initializations to increase reliability. The overall performance is reported as the average across these three runs.

### *4.4. Evaluation Measures*

We employ two performance metrics to evaluate the effectiveness of the methods: the average area under the precision-recall curve (AUPRC) and the ranking loss measure. Given that the tasks involve multi-label classification and hierarchical multi-label classification, we use a variant of AUPRC–the area under the micro-averaged Precision-Recall curve (AU$\overline{\text{PRC}}$). These metrics are chosen because they are independent of classification thresholds and provide a reliable assessment of performance.

In this context, $\overline{\text{Prec}}$ represents the proportion of predicted labels that are correct, while $\overline{\text{Rec}}$ corresponds to the proportion of actual labels in the dataset that are correctly predicted. These values are calculated as follows:

$$\overline{\text{Prec}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FP}_i}, \tag{10}$$

$$\overline{\text{Rec}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FN}_i}, \tag{11}$$

where $i$ iterates over all classes. By varying the decision threshold, an average Precision-Recall (PR) curve is generated.

Ranking loss (RL), measuring the average fraction of incorrectly ordered label pairs for a given example, is defined as

Table 2: HELM variants and baseline methods with their loss component configurations for systematic evaluation.

| **Method Variant** | **Notation** | $\mathcal{L}_s$ | $\mathcal{L}_g$ | $\mathcal{L}_b$ | **Task** |
|---|---|---|---|---|---|
| MLC Baseline | MLC | ✓ | | | {SL} |
| HMLC Baseline | HMLC | ✓ | | | {SL} |
| HELM (Graph-only) | $\text{HELM}_g$ | ✓ | ✓ | | {SL, SSL} |
| HELM (BYOL-only) | $\text{HELM}_b$ | ✓ | | ✓ | {SL, SSL} |
| HELM (Full) | HELM | ✓ | ✓ | ✓ | {SL, SSL} |

$$\text{RL} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i^+|\cdot|Y_i^-|}\sum_{y_p\in Y_i^+}\sum_{y_n\in Y_i^-}\mathbb{I}(f_i(y_p)\le f_i(y_n)), \tag{12}$$

where $N$ is the number of examples, $Y_i^+$ and $Y_i^-$ are the sets of positive and negative labels for the $i$-th example, $f_i(y)$ is the predicted score for label $y$, and $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise. The loss is weighted by the sizes of $Y_i^+$ and $Y_i^-$, penalizing each pair of misordered labels equally. A lower ranking loss indicates better performance, as fewer label pairs are incorrectly ranked.

To compute these metrics, we use the readily available implementations from `scikit-learn` [39].

### *4.5. Methods for Comparison*

#### *4.5.1. HELM Variants and Baseline Methods*

To systematically evaluate the contribution of each component, we analyze different variants of HELM by selectively including loss terms from the overall objective (Eq. 9). Table 2 summarizes all evaluated configurations and corresponding loss components.

We establish two baseline methods to isolate the impact of hierarchical information: *(i)* the MLC baseline, which uses only leaf labels during training and performs standard multi-label classification, and *(ii)* the HMLC baseline, which incorporates the complete hierarchical label set during training but treats all labels independently without exploiting structural relationships (essentially a flat approach to hierarchical classification). Both baselines utilize only the supervised classification loss $\mathcal{L}_s$, omitting the graph and self-supervised components.

The HELM variants systematically introduce additional components to assess their individual and combined contributions. $\text{HELM}_g$ adds graph-based dependency modeling to capture hierarchical label relationships, while $\text{HELM}_b$ incorporates self-supervised learning through BYOL to leverage unlabeled data. The complete HELM model integrates all three components, providing a unified framework that combines supervised classification, hierarchical reasoning, and self-supervised representation learning.

This experimental design enables precise measurement of how each component contributes to performance, particularly in scenarios with limited labeled supervision. All methods are evaluated using exclusively leaf labels of testing examples to ensure fair comparison, regardless of training configuration.

#### *4.5.2. State-of-the-Art Methods*

We also compare HELM against three state-of-the-art HMLC methods: C-HMCNN [15], HiMulConE [6], and HMI [40]. These methods represent a diverse range of approaches to hierarchical multi-label classification, highlighting the strengths and limitations of current techniques.

C-HMCNN [15] ensures consistency between parent and child predictions through a modified binary cross-entropy (BCE) loss, providing flexibility in handling complex hierarchies. Although originally designed for tabular

data, we opted to use C-HMCNN due to the scarcity of native HMLC methods for images. To integrate C-HMCNN into our evaluation setup, we adapted it for image-based tasks by replacing its tabular encoder with our vision transformer encoder. HiMulConE [6] adopts a two-stage approach: it first employs a hierarchy-preserving contrastive loss to learn label-aware embeddings and then trains a classifier on leaf nodes using cross-entropy and softmax. Since the original method utilizes a cross-entropy loss designed for single-label classification at the leaf level, we replaced it with a BCE loss to support scenarios where multiple labels may exist at the leaf level. HMI [40] utilizes a hyperbolic Poincaré ball model to encode logical relationships, such as implication and exclusion, through geometric constraints. This approach achieves high consistency with fewer dimensions, enhancing computational efficiency. However, similar to C-HMCNN, HMI was originally designed for tabular data: We adapted it for image-based tasks by incorporating our vision transformer encoder to enable a direct comparison.

For a fair evaluation, all methods are implemented using our vision encoder, and we utilize the official code repositories of each method, setting hyperparameters according to the original authors' recommendations.

Table 3: Comparison of HELM and its variants against the MLC and HMLC baseline methods, all trained in a supervised learning setting. The results present the AUPRC and ranking loss on test data, averaged across three runs. For each dataset, the best performance is highlighted in **bold**, while the second-best is underlined. Relative improvements are shown as percentages in parentheses (vs MLC, vs HMLC), with positive values highlighted in blue and negative values in red.

| Method | Remote Sensing Datasets | | | | Rank |
|---|---|---|---|---|---|
| | UCM | AID | DFC-15 | MLRSNet | |
| | | | **AUPRC (↑)** | | |
| MLC | 0.863 | 0.767 | 0.967 | 0.838 | 5.00 |
| HMLC | 0.890 | 0.827 | 0.971 | 0.863 | 3.00 |
| $\text{HELM}_g$ | 0.899 (+4.2, +1.0) | 0.842 (+9.8, +1.8) | **0.979** (+1.2, +0.8) | 0.869 (+3.7, +0.7) | 2.50 |
| $\text{HELM}_b$ | 0.885 (+2.5, -0.6) | **0.852** (+11.1, +3.0) | 0.969 (+0.2, -0.2) | **0.873** (+4.2, +1.2) | 3.13 |
| HELM | **0.904** (+4.8, +1.6) | 0.849 (+10.7, +2.7) | 0.977 (+1.0, +0.6) | 0.871 (+3.9, +0.9) | **1.38** |
| | | | **Ranking Loss (↓)** | | |
| MLC | 0.031 | 0.025 | 0.010 | 0.039 | 4.75 |
| HMLC | 0.031 | 0.021 | 0.008 | 0.027 | 2.88 |
| $\text{HELM}_g$ | 0.024 (+22.6, +22.6) | 0.019 (+24.0, +9.5) | 0.007 (+30.0, +12.5) | 0.025 (+35.9, +7.4) | 2.31 |
| $\text{HELM}_b$ | 0.029 (+6.5, +6.5) | 0.019 (+24.0, +9.5) | 0.012 (-20.0, -50.0) | 0.025 (+35.9, +7.4) | 3.75 |
| HELM | **0.022** (+29.0, +29.0) | **0.017** (+32.0, +19.0) | **0.006** (+40.0, +25.0) | **0.024** (+38.5, +11.1) | **1.31** |
| **Method** | **Medical Imaging Datasets** | | | | **Rank** |
| | **MuRed** | **HPA** | **ChestX-14** | **PadChest** | |
| | | | **AUPRC (↑)** | | |
| MLC | 0.486 | 0.443 | 0.197 | 0.106 | 5.00 |
| HMLC | 0.598 | 0.527 | 0.261 | 0.137 | 3.00 |
| $\text{HELM}_g$ | 0.589 (+21.2, -1.5) | 0.516 (+16.5, -2.1) | 0.256 (+30.0, -1.9) | 0.145 (+36.8, +5.8) | 2.50 |
| $\text{HELM}_b$ | 0.497 (+2.3, -16.9) | 0.469 (+5.9, -11.0) | 0.239 (+21.3, -8.4) | 0.140 (+32.1, +2.2) | 3.13 |
| HELM | **0.621** (+27.8, +3.8) | **0.536** (+21.0, +1.7) | **0.266** (+35.0, +1.9) | **0.147** (+38.7, +7.3) | **1.38** |
| | | | **Ranking Loss (↓)** | | |
| MLC | 0.104 | 0.055 | 0.163 | 0.043 | 4.75 |
| HMLC | 0.082 | 0.051 | 0.121 | 0.035 | 2.88 |
| $\text{HELM}_g$ | 0.090 (+13.5, -9.8) | 0.050 (+9.1, +2.0) | 0.121 (+25.8, +0.0) | **0.034** (+20.9, +2.9) | 2.31 |
| $\text{HELM}_b$ | 0.102 (+1.9, -24.4) | 0.058 (-5.5, -13.7) | 0.125 (+23.3, -3.3) | 0.037 (+14.0, -5.7) | 3.75 |
| HELM | **0.064** (+38.5, +22.0) | **0.046** (+16.4, +9.8) | **0.119** (+27.0, +1.7) | 0.036 (+16.3, -2.9) | **1.31** |

## 5. Results

This section presents a comprehensive evaluation of our HELM framework across eight benchmark datasets from remote sensing and medical imaging domains under both supervised and semi-supervised learning settings.

Through experiments, we aim to answer the following research questions:

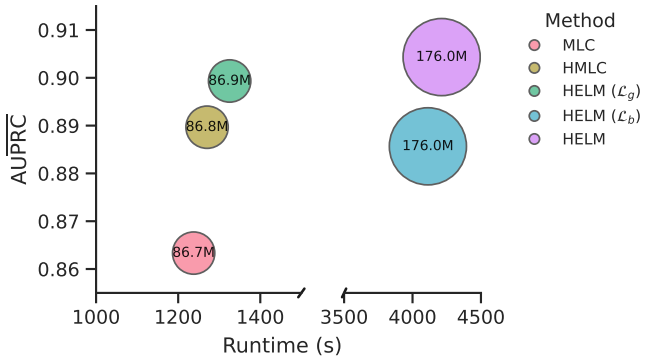



Figure 4: Comparison of training times, performance, and parameter counts for the baseline methods and HELM on the UCM dataset. The size of each bubble corresponds to the number of parameters (in millions).

i Does incorporation of the label hierarchy yield models that lead to better predictive performance?

ii What is the contribution of individual components in the proposed hierarchy-informed loss?

iii Does the addition of unlabeled data improve the predictive performance for the hierarchy-informed models in semi-supervised setting?

iv How do the hierarchy-informed models compare to the state-of-the-art methods?

v Do hierarchy-informed models learn more informative label space embeddings?

### *5.1. Comparison to Supervised Baselines*

Table 3 presents the performance comparison between the complete HELM model, individual components (HELM$_g$ for graph learning; HELM$_b$ for self-supervised learning), and MLC and hierarchical HMLC baselines. The results reveal that HELM consistently outperforms all baseline methods across both remote sensing and medical imaging datasets. The HMLC baseline also consistently outperforms the MLC baseline.

The HELM model achieves the highest performance in 5 out of 8 datasets for $\overline{\text{AUPRC}}$ and demonstrates superior ranking loss performance across 7 datasets. Notably, HELM attains the top average rank of 1.38 for $\overline{\text{AUPRC}}$ and 1.31 for ranking loss. In sum, HELM shows robust performance across diverse HMLC scenarios.

#### *5.1.1. Component Contribution Analysis*

To understand the individual contributions of our framework components, we analyze the performance of HELM$_g$ (graph learning only) and HELM$_b$ (self-supervised learning only) against the complete HELM model.

The graph learning component ($\mathcal{L}_g$) emerges as the most impactful individual component, consistently improving performance over the MLC baseline and often exceeding the HMLC baseline. The self-supervised component ($\mathcal{L}_b$) demonstrates somewhat domain-specific effectiveness, yielding more notable improvements on RSI datasets. It achieves substantial gains in $\overline{\text{AUPRC}}$ on AID (0.852) and MLRSNet (0.873), effectively leveraging semantic visual structures through contrastive learning. However, performance varies in medical imaging contexts, where pathological-anatomical relationships seem to present distinct challenges.

The complete HELM model demonstrates synergistic benefits through the integration of both components, including 27.8% improvement on MuRed and 35.0% improvement on ChestX-14 over the MLC baseline. This

performance stems from the complementary strengths of graph-based hierarchical learning and self-supervised representation learning, which create unified learning mechanisms that effectively balance hierarchical reasoning with visual representation learning.

The ablation study confirms that while individual components offer distinct advantages, their integration creates synergistic effects that surpass the sum of their parts, demonstrating the ability of our unified approach to successfully exploit hierarchical dependencies among classes.

### *5.1.2. Computational Efficiency Analysis*

Figure 4 illustrates the computational trade-offs inherent in our framework design, evaluated on the representative UCM dataset. The graph learning component proves remarkably efficient, adding only 107K parameters while delivering substantial performance gains. In contrast, the BYOL component introduces significant computational overhead due to its dual-encoder architecture.

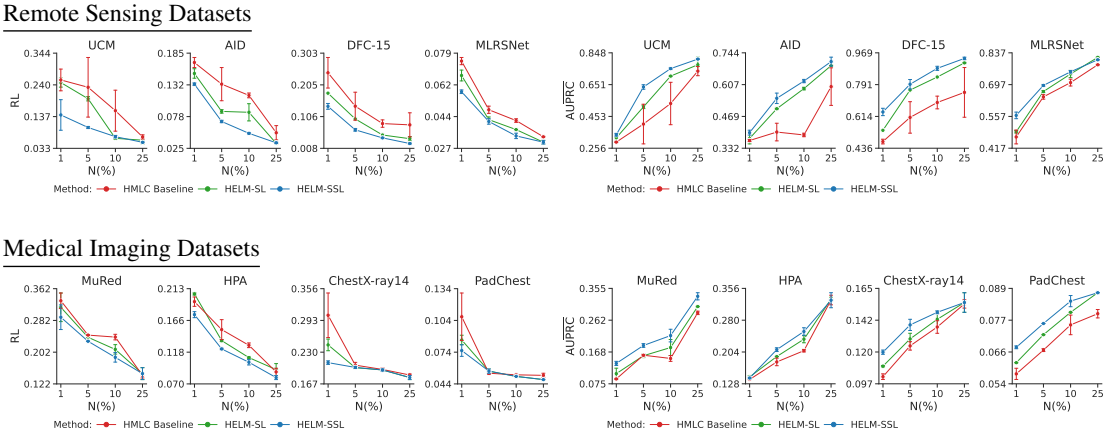



Figure 5: Comparison of HELM's predictive performance in supervised (HELM-SL) and semi-supervised (HELM-SSL) configurations against the HMLC baseline across varying proportions of labeled data. Results are shown for remote sensing datasets with AUPRC and ranking loss (RL) in the top row, and for medical image datasets with AUPRC and RL in the bottom row. The learning curves present the mean ($\mu$) and standard deviation ($\sigma$) across three runs. Higher AUPRC values and lower RL values indicate better predictive performance. HELM-SSL consistently outperforms both HELM-SL and the baseline, with the most pronounced improvements observed in low-resource scenarios.

## *5.2. Semi-Supervised Learning Analysis*

Having established HELM's effectiveness in supervised settings, we now investigate its capability to leverage unlabeled data through semi-supervised learning. We examine HELM's performance across varying proportions of labeled data (1%, 5%, 10%, and 25%) to understand how effectively the framework exploits unlabeled instances for predictive performance enhancement.

We compare HELM's performance in both supervised (HELM-SL) and semi-supervised (HELM-SSL) configurations against the supervised HMLC baseline. Both configurations use all three (classification, graph learning, and self-supervised) branches, but differ in data utilization. In HELM-SL, the classification branch operates exclusively on labeled samples, while the graph branch and self-supervised branch process only labeled data for both feature learning and loss computation. The self-supervised branch treats the labeled batch as unlabeled data by discarding the labels during training. In HELM-SSL, the classification branch continues to use only labeled samples, while the graph branch processes the entire batch (labeled and unlabeled) for feature refinement but com-

putes loss exclusively on labeled samples. The self-supervised branch treats the entire batch as unlabeled data to learn robust visual representations.

### *5.2.1. Learning Curves*

The learning curves depicted in Figure 5 demonstrate HELM's consistent improvements over the supervised HMLC baseline across both configurations. Notably, HELM-SSL consistently outperforms HELM-SL, highlighting the effectiveness of incorporating unlabeled data through the BYOL component. The performance gap between HELM-SSL and HELM-SL is most substantial when labeled data is extremely scarce (1-5%): In this case, the self-supervised learning component successfully leverages unlabeled instances to enhance representation quality and maintain robust predictive performance.

This capability underscores HELM's value in real-world applications where obtaining labeled data is expensive or time-consuming, demonstrating that the framework can effectively exploit abundant unlabeled data to achieve superior performance even with minimal supervision.

### *5.2.2. Component Contribution Analysis*

Table 4: Evaluation of HELM's individual loss components in SSL setting compared to the supervised HMLC baseline. Models are trained with varying proportions of labeled data and evaluated on a fixed test set. Results are reported using $\overline{\text{AUPRC}}$ (higher values indicate better performance) and Ranking Loss (lower values indicate better performance), averaged over three runs. The best performance for each dataset and labeled data proportion is highlighted in bold.

(a) Remote Sensing Datasets

| $\overline{\text{AUPRC}}(\uparrow)$ | UCM | | | | AID | | | | DFC-15 | | | | MLRSNet | | | | Avg. ranks | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N (%) | | | | N (%) | | | | N (%) | | | | N (%) | | | | N (%) | | | |
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| HMLC Baseline | 0.268 | 0.357 | 0.479 | 0.691 | 0.366 | 0.410 | 0.382 | 0.564 | 0.481 | 0.619 | 0.692 | 0.766 | 0.470 | 0.643 | 0.716 | 0.787 | 3.3 | 3.8 | 3.5 | 3.8 |
| HELM$_b$ | 0.290 | 0.554 | 0.656 | 0.770 | 0.362 | 0.461 | 0.557 | 0.671 | 0.522 | 0.717 | 0.824 | 0.909 | 0.380 | 0.643 | 0.709 | 0.788 | 3.3 | 3.0 | 3.0 | 2.8 |
| HELM$_g$ | 0.269 | 0.532 | 0.714 | 0.799 | 0.375 | 0.485 | 0.597 | 0.680 | 0.619 | 0.792 | 0.875 | 0.916 | 0.422 | 0.657 | 0.707 | 0.781 | 2.5 | 2.0 | 2.5 | 2.5 |
| HELM | **0.335** | **0.651** | **0.730** | **0.807** | **0.390** | **0.533** | **0.631** | **0.701** | **0.659** | **0.790** | **0.885** | **0.934** | **0.557** | **0.666** | **0.743** | **0.816** | **1.0** | **1.3** | **1.0** | **1.0** |
| RL(↓) | UCM | | | | AID | | | | DFC-15 | | | | MLRSNet | | | | Avg. ranks | | | |
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| HMLC Baseline | 0.266 | 0.232 | 0.178 | 0.080 | 0.169 | 0.122 | 0.137 | 0.062 | 0.232 | 0.164 | 0.101 | 0.088 | 0.074 | 0.048 | 0.042 | 0.033 | 3.0 | 3.8 | 3.8 | 3.5 |
| HELM$_b$ | 0.247 | 0.125 | 0.086 | 0.057 | 0.216 | 0.088 | 0.059 | 0.041 | 0.179 | 0.092 | 0.058 | 0.033 | 0.098 | 0.051 | 0.043 | 0.034 | 2.3 | 2.5 | 2.5 | 2.8 |
| HELM$_g$ | 0.307 | 0.127 | 0.068 | 0.050 | 0.155 | 0.076 | 0.056 | 0.039 | 0.137 | 0.067 | 0.039 | 0.031 | 0.084 | 0.043 | 0.041 | 0.033 | 3.8 | 2.3 | 2.3 | 2.0 |
| HELM | **0.137** | **0.106** | **0.072** | **0.051** | **0.133** | **0.071** | **0.048** | **0.033** | **0.132** | **0.065** | **0.037** | **0.023** | **0.061** | **0.041** | **0.036** | **0.030** | **1.0** | **1.3** | **1.3** | **1.3** |

(b) Medical Imaging Datasets

| $\overline{\text{AUPRC}}(\uparrow)$ | MuRed | | | | HPA | | | | ChestXray-14 | | | | PadChest | | | | Avg. ranks | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N (%) | | | | N (%) | | | | N (%) | | | | N (%) | | | | N (%) | | | |
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| HMLC Baseline | 0.093 | 0.160 | 0.167 | 0.281 | 0.141 | 0.181 | 0.217 | 0.315 | 0.101 | 0.125 | 0.141 | 0.155 | 0.057 | 0.065 | 0.072 | 0.079 | 3.5 | 3.8 | 3.8 | 3.0 |
| HELM$_b$ | 0.112 | 0.165 | 0.208 | 0.282 | 0.134 | 0.178 | 0.227 | 0.269 | 0.116 | 0.129 | 0.139 | 0.152 | 0.064 | 0.073 | 0.080 | 0.083 | 3.3 | 3.3 | 3.0 | 3.0 |
| HELM$_g$ | 0.115 | 0.179 | 0.222 | 0.304 | 0.139 | 0.192 | 0.240 | 0.313 | 0.121 | 0.134 | 0.142 | 0.151 | 0.066 | 0.074 | 0.079 | 0.083 | 2.0 | 2.0 | 2.3 | 3.0 |
| HELM | **0.133** | **0.189** | **0.226** | **0.312** | **0.145** | **0.209** | **0.245** | **0.323** | **0.120** | **0.138** | **0.146** | **0.158** | **0.068** | **0.077** | **0.083** | **0.086** | **1.3** | **1.0** | **1.0** | **1.0** |
| RL(↓) | MuRed | | | | HPA | | | | ChestXray-14 | | | | PadChest | | | | Avg. ranks | | | |
| Methods | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 | 1 | 5 | 10 | 25 |
| HMLC Baseline | 0.357 | 0.250 | 0.222 | 0.147 | 0.189 | 0.147 | 0.126 | 0.093 | 0.317 | 0.202 | 0.192 | 0.184 | 0.116 | 0.057 | 0.053 | 0.052 | 3.3 | 2.8 | 3.3 | 2.8 |
| HELM$_b$ | 0.311 | 0.237 | 0.201 | 0.161 | 0.234 | 0.143 | 0.119 | 0.109 | 0.241 | 0.207 | 0.202 | 0.186 | 0.086 | 0.062 | 0.053 | 0.051 | 2.8 | 2.8 | 3.0 | 3.8 |
| HELM$_g$ | 0.345 | 0.252 | 0.186 | 0.144 | 0.220 | 0.149 | 0.119 | 0.092 | 0.222 | 0.197 | 0.189 | 0.185 | 0.083 | 0.057 | 0.055 | 0.049 | 2.5 | 3.0 | 2.5 | 1.8 |
| HELM | **0.317** | **0.238** | **0.178** | **0.155** | **0.199** | **0.134** | **0.113** | **0.092** | **0.215** | **0.197** | **0.191** | **0.178** | **0.074** | **0.055** | **0.050** | **0.048** | **1.5** | **1.5** | **1.3** | **1.8** |

To understand the individual contributions of HELM's components in the SSL setting, we evaluate each loss term's impact on performance across varying labeled data proportions using the HELM-SSL configuration (referred to as HELM for notation simplicity). Table 4 presents comprehensive comparisons between HELM variants and the supervised baseline across remote sensing (a) and medical imaging (b) domains.

The analysis reveals that the graph learning component ($\mathcal{L}_g$) provides the most substantial improvements, particularly excelling in scenarios with moderate labeled data availability. This demonstrates that hierarchical

label dependencies enhance predictive performance even in SSL settings, where the graph structure effectively propagates information from labeled to unlabeled instances.

The BYOL component ($\mathcal{L}_b$) demonstrates complementary benefits, with its contribution being most pronounced in extremely low-resource settings where self-supervised representation learning becomes crucial.

The synergistic effect of combining both components is evident in the full HELM model, which consistently achieves the best performance across all experimental configurations. This demonstrates that the graph learning and self-supervised components address different aspects of the multi-label learning problem, with their integration yielding superior results compared to either component in isolation.

### *5.3. Comparison to State-of-the-Art Methods*

Table 5: Comparison of HELM's predictive performance with state-of-the-art methods on supervised HMLC tasks. Results are presented using the $\overline{\text{AUPRC}}$ metric (higher values indicate better performance) and ranking loss (lower values indicate better performance), averaged over three runs on test data. Best in **bold**, second best underlined.

| | **Remote Sensing Datasets** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **AUPRC (↑)** | | | | **Ranking Loss (↓)** | | | |
| **Method** | **UCM** | **AID** | **DFC-15** | **MLRSNet** | **UCM** | **AID** | **DFC-15** | **MLRSNet** |
| C-HMCNN [15] | 0.834 | 0.764 | 0.962 | 0.792 | 0.038 | 0.024 | 0.012 | 0.041 |
| HiMulConE [6] | 0.843 | 0.770 | 0.970 | 0.865 | 0.031 | 0.020 | **0.006** | 0.035 |
| HMI [40] | 0.661 | 0.647 | 0.923 | 0.437 | 0.080 | 0.073 | 0.043 | 0.138 |
| HELM (Ours) | **0.904** | **0.849** | **0.977** | **0.871** | **0.022** | **0.017** | **0.006** | **0.025** |
| | **Medical Imaging Datasets** | | | | | | | |
| | **AUPRC (↑)** | | | | **Ranking Loss (↓)** | | | |
| **Method** | **MuRed** | **HPA** | **ChestX-ray14** | **PadChest** | **MuRed** | **HPA** | **ChestX-ray14** | **PadChest** |
| C-HMCNN [15] | 0.441 | 0.459 | 0.194 | 0.100 | 0.123 | 0.056 | 0.168 | 0.046 |
| HiMulConE [6] | 0.466 | 0.496 | 0.189 | 0.102 | 0.107 | **0.044** | 0.167 | 0.046 |
| HMI [40] | 0.246 | 0.102 | 0.107 | 0.056 | 0.239 | 0.296 | 0.205 | 0.073 |
| HELM (Ours) | **0.621** | **0.536** | **0.266** | **0.147** | **0.064** | 0.046 | **0.119** | **0.036** |

In this section, we compare HELM with three state-of-the-art HMLC methods: C-HMCNN [15], HiMulConE [6], and HMI [40]. Given that these are supervised methods, HELM is also evaluated in a supervised setting to ensure a fair comparison.

Table 5 presents the results, averaged over three runs on the test data. HELM consistently outperforms the state-of-the-art methods across datasets. Specifically, when considering $\overline{\text{AUPRC}}$, HELM achieves the best performance across all eight datasets, often surpassing the second-best method by a large margin. For example, on MuRed, ChestX-ray14, and PadChest datasets, HELM improves between 33% and 44% over HiMulConE, the second best competitor. Regarding ranking loss, HELM achieves the best result in seven datasets, while on the remaining dataset, it is a close second best after the HiMulConE method. Notably, among the competitors, HiMulConE is consistently the second best, while HMI consistently delivers the lowest performance across both metrics.

Comparison with state-of-the-art methods demonstrated HELM's ability to effectively exploit hierarchical label constraints and achieve robust performance across diverse datasets, distinguishing it from other approaches dealing with the HMLC task.

### *5.4. Qualitative evaluation*

This section presents a qualitative evaluation of the proposed HELM method compared to the state-of-the-art methods from the literature. For simplicity of analysis, we used the UCM dataset.

*5.4.1. Visualization of the learned embeddings*

The embeddings are derived from the hierarchy-specific `CLS` tokens ($T_h$) produced by the final layer of the encoder. Since our encoder serves as a common backbone architecture across state-of-the-art methods, the comparison between the learned embeddings is both fair and direct, ensuring that differences arise solely from the design and learning mechanisms of the methods. Additionally, we compare HELM before and after introducing the graph component to evaluate the impact of the hierarchical structure on the quality of the learned embeddings. To visualize the high-dimensional embeddings learned by the methods, we use the Uniform Manifold Approximation and Projection (UMAP).

To quantify the quality of the embeddings, we use the Normalized Mutual Information (NMI) measure, which evaluates the alignment between the clusters formed by embeddings and the ground-truth labels. Before computing NMI, we apply a $k$-Nearest Neighbors ($k$-NN) clustering approach on the original embeddings, with the number of clusters set to match the number of labels at each hierarchical level. Additionally, we report the average NMI, calculated by averaging the NMI values across different levels of the hierarchy. This approach provides a comprehensive evaluation of both fine-grained (leaf-level) and broader (ancestor-level) clustering quality, as well as the overall quality of the embeddings, ensuring the learned representations effectively capture the hierarchical structure of the labels.

Figure 6 presents the UMAP analysis results across the hierarchical levels of the UCM dataset, showcasing the performance of various methods. HELM with hierarchical modeling delivers the best results, achieving NMI values of 0.411 at the first level and 0.801 at the second level of the hierarchy, with an average NMI of $\overline{\text{NMI}} = 0.737$ across all levels. The embeddings produced are well-structured and form distinct clusters that align with the hierarchical label relationships, demonstrating the effectiveness of explicitly modeling hierarchical dependencies. This approach allows the model to capture both coarse-grained and fine-grained relationships, resulting in superior clustering quality and predictive performance. In contrast, the HMLC baseline achieves $\overline{\text{NMI}}$ of 0.612, placing it as the second-lowest performing method in the comparison. This highlights the significant limitations of excluding hierarchical relationships, which restrict the model's ability to organize embeddings meaningfully. While HiMulConE delivers competitive results, it falls short of capturing the hierarchical structure as effectively as HELM with hierarchical modeling, further emphasizing the importance of integrating graph-based reasoning to incorporate hierarchical relationships for improved alignment and performance.

*5.4.2. Performance on image retrieval*

To assess whether HELM has learned meaningful and discriminative representations, we conduct a content-based image retrieval experiment using the $k$-NN algorithm. This approach allows us to evaluate the model's ability to group and distinguish semantically relevant images based on their learned representations. For this experiment, we set $k = 5$, retrieving the top-5 most representative images for a given query image. The retrieved images are sorted in ascending order of their distance from the query image, highlighting the closest matches.

Figure 7 illustrates the results of the image retrieval experiment, highlighting the performance of HELM and the state-of-the-art methods. In Figure 7 (a), HELM demonstrates exceptional performance in retrieving semantically consistent images in complex scenes with a higher number of labels, showcasing its ability to effectively leverage label correlations. In less complex scenes with fewer labels, as depicted in Figure 7 (b), all methods perform competitively, including both variants of HELM. Analyzing the performance of other methods, C-HMCNN

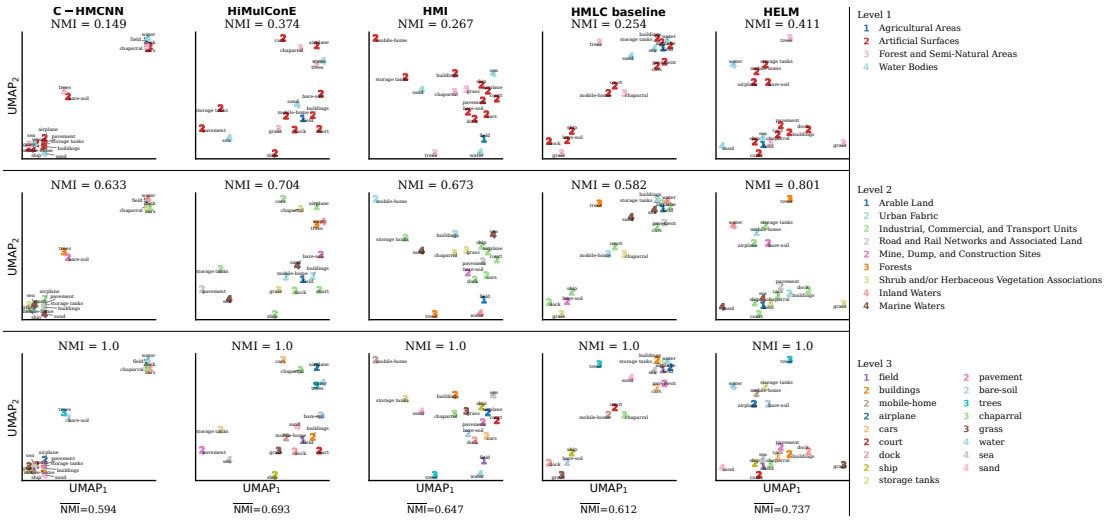



Figure 6: Comparison of the 2-D UMAP embeddings between HELM and the state-of-the-art methods for the UCM dataset. The learned embeddings are colored based on different levels of the UCM label hierarchy. The visualization is based on the embeddings corresponding to the leaf labels, while the color coding reflects the grouping and relationships at each hierarchical level. The NMI values are reported for each method, where higher values indicate better alignment between the clusters and the ground truth labels, reflecting the quality of the hierarchical embeddings.

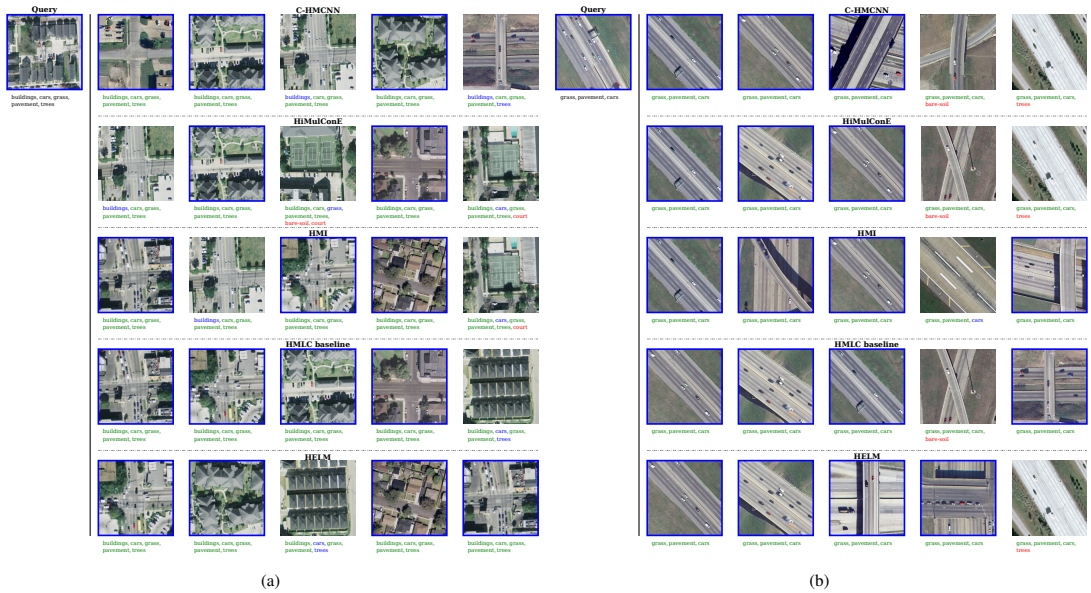



Figure 7: Top-5 retrieved images for the query image, showcasing a comparison of HELM against state-of-the-art methods. The query image is displayed on the left, followed by results from C-HMCNN, HiMulConE, HMI, HMLC baseline, and HELM. Labels in green denote exact matches, blue denotes the query's semantic labels, and red indicates mismatched labels. The images with blue borders indicate exact matches between the query's semantic labels and those of the retrieved images.

and HiMulConE (despite their strong predictive performance in the downstream MLC task (see Table 5)) struggle to enforce semantic consistency at fine-grained levels, leading to mismatched retrievals in more complex label scenarios. HMI, while showing improved semantic alignment in content-based image retrieval, performs poorly in the MLC task.

### *5.5. Limitations*

The results presented so far demonstrate that HELM outperforms state-of-the-art models from the literature. However, our approach is not without limitations. One drawback is the computationally intensive BYOL branch (as shown in Figure 4), which, while contributing to improved performance, comes at the cost of significantly increased runtime and parameter count due to its dual-encoder architecture. This is, arguably, justified by the enhanced representation quality and improved generalization capabilities observed across our experimental evaluation.

Furthermore, as shown in the UMAP analysis (Figure 6), while HELM outperforms state-of-the-art methods and generates well-defined clusters that align with the hierarchical structure of the labels, it occasionally fails to consistently cluster hierarchical levels. For instance, labels such as "`sand`" and "`water`", which share the same parent label "`Water Bodies`" at higher levels of the hierarchy, are well-clustered individually but fail to fully reflect their shared broader grouping. This suggests that although HELM excels at modeling finer-grained relationships, there is room for improvement in capturing and representing broader hierarchical structures more consistently.

## 6. Conclusion

In this paper, we introduced HELM, a novel HMLC framework designed to capture complex relationships among labels, particularly in scenarios where images belong to multiple paths within a class hierarchy. The framework integrates hierarchy-specific class tokens to capture internal label interactions and leverages a graph-based structure encoder, enabling HELM to learn hierarchy-aware embeddings that preserve label relationships while enhancing the discriminative power of the representations. Furthermore, HELM incorporates BYOL as a self-supervised learning module, allowing it to leverage unlabeled data in addition to limited labeled data. This design not only boosts representation quality but also makes HELM well-suited for semi-supervised learning tasks, making it (to the best of our knowledge) the only semi-supervised HMLC method for images.

Through rigorous experiments on eight datasets – four from the remote sensing domain and four from the medical domain – we show that incorporating hierarchical information significantly improves predictive performance, yielding consistent gains over baseline methods for both MLC and HMLC in supervised and semi-supervised settings. Furthermore, HELM outperforms the state-of-the-art methods, showcasing its effectiveness in tackling the challenges of HMLC across diverse and complex datasets. Finally, for the purposes of the evaluation we conducted, we constructed label hierarchies for eight datasets to enrich the image classification domain, where such resources are scarce.

Future work will focus on developing end-to-end hierarchy learning methods that automatically discover taxonomic structures directly from visual data, reducing reliance on external knowledge while maintaining structured learning benefits. Additionally, we plan to investigate visual-semantic initialization strategies that better capture

the alignment between visual appearance and hierarchical structure, potentially using vision-language models trained specifically on hierarchical visual data to create more effective starting embeddings that reflect both semantic meaning and visual similarity.

**Declaration of Generative AI and AI-assisted technologies in the writing process**

During the preparation of this work the author(s) used ChatGPT in order to paraphrase and reformulate some sentences based on authors' initial draft. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

**Acknowledgment**

We acknowledge the support of the European Commission via the projects ASSAS (grant number 101059682), ELIAS (101120237), INQUIRE (101057499) and PARC (101057014), as well as by the Slovenian Research Agency via the research program P2-0103 and the research projects J1-3033, J2-2505, J2-4452, J2-4660, J3-3070, J4-3095, J5-4575, J7-4636, J7-4637, and N2-0236. The work of the BK was supported by the Young Researcher grant PR-12394.

**References**

[1] Ping Fu Shou Feng and Wenbin Zheng. A hierarchical multi-label classification method based on neural networks for gene function prediction. *Biotechnology & Biotechnological Equipment*, 32(6):1613–1621, 2018.

[2] Jibing Gong, Zhiyong Teng, Qi Teng, Hekai Zhang, Linfeng Du, Shuai Chen, Md Zakirul Alam Bhuiyan, Jianhua Li, Mingsheng Liu, and Hongyuan Ma. Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. *IEEE Access*, 8:30885–30896, 2020.

[3] Haomin Chen, Shun Miao, Daguang Xu, Gregory D. Hager, and Adam P. Harrison. Deep hiearchical multi-label classification applied to chest x-ray abnormality taxonomies. *Medical Image Analysis*, 66:101811, 2020.

[4] Jurica Levatić, Dragi Kocev, Marko Debeljak, and Sašo Džeroski. Community structure models are improved by exploiting taxonomic rank with predictive clustering trees. *Ecological modelling*, 306:294–304, 2015.

[5] Jurica Levatić, Dragi Kocev, and Sašo Džeroski. The importance of the label hierarchy in hierarchical multi-label classification. *Journal of Intelligent Information Systems*, 45(2):247–271, October 2015.

[6] Shu Zhang, Ran Xu, Caiming Xiong, and Chetan Ramaiah. Use all the labels: A hierarchical multi-label contrastive learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16660–16669, June 2022.

[7] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[8] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[9] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[10] Jian Kang, Rubén Fernández-Beltrán, Zhen Ye, Xiaohua Tong, Pedram Ghamisi, and Antonio Plaza. High-rankness regularized semi-supervised deep metric learning for remote sensing imagery. *Remote Sensing*, 12(16), 2020.

[11] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical classification of diatom images using ensembles of predictive clustering trees. *Ecological Informatics*, 7(1):19–29, 2012.

[12] Brendan Kolisnik, Isaac Hogan, and Farhana Zulkernine. Condition-CNN: A hierarchical multi-label fashion image classification model. *Expert Systems with Applications*, 182:115195, 2021.

[13] Xinqi Zhu and Michael Bain. B-cnn: branch convolutional neural network for hierarchical classification. *arXiv preprint arXiv:1709.09890*, 2017.

[14] Khondaker Tasrif Noor and Antonio Robles-Kelly. H-capsnet: A capsule network for hierarchical image classification. *Pattern Recognition*, 147:110135, 2024.

[15] Eleonora Giunchiglia and Thomas Lukasiewicz. Coherent hierarchical multi-label classification networks. *Advances in neural information processing systems*, 33:9662–9673, 2020.

[16] Jingzhou Chen, Peng Wang, Jian Liu, and Yuntao Qian. Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4858–4867, 2022.

[17] Jingzhou Chen and Yuntao Qian. Hierarchical multilabel ship classification in remote sensing images using label relation graphs. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022.

[18] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019.

[19] Ziwen Lan, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Hierarchical multi-label attribute classification with graph convolutional networks on anime illustration. *IEEE Access*, 11:35447–35456, 2023.

[20] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[21] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[22] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, page 92–100, New York, NY, USA, 1998. Association for Computing Machinery.

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, June 2022.

[24] Marjan Stoimchev, Jurica Levatić, Dragi Kocev, and Sašo Džeroski. Semi-supervised multi-label classification of land use/land cover in remote sensing images with predictive clustering trees and ensembles. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024.

[25] Junfei Xiao, Yutong Bai, Alan Yuille, and Zongwei Zhou. Delving into masked autoencoders for multi-label thorax disease classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3588–3600, January 2023.

[26] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[27] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone. "Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):1144–1158, 2018.

[28] Y. Hua, L. Mou, and X.X. Zhu. Relation Network for Multi-label Aerial Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[29] Y. Hua, L. Mou, and X.X. Zhu. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:188–199, 2019.

[30] Q. Xiaoman Qi, Z. Panpan, W. Yuebin, Z. Liqiang, P. Junhuan, W. Mengfan, C. Jialong, Z. Xudong, Z. Ning, and P.M.Takis. MLRSNet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:337–350, 2020.

[31] Manuel Alejandro Rodríguez, Hasan AlMarzouqi, and Panos Liatsis. Multi-label retinal disease classification using transformers. *IEEE Journal of Biomedical and Health Informatics*, 27(6):2739–2750, 2022.

[32] Wei Ouyang, Casper F Winsnes, Martin Hjelmare, Anthony J Cesnik, Lovisa Åkesson, Hao Xu, Devin P Sullivan, Shubin Dai, Jun Lan, Park Jinmo, et al. Analysis of the human protein atlas image classification competition. *Nature methods*, 16(12):1254–1261, 2019.

[33] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical Image Analysis*, 66:101797, 2020.

[34] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[35] Referans Copernicus. Corine land cover. *Copernicus Land Monitoring Service. L. Monit. Serv*, 2018.

[36] World Health Organization et al. The icd-10 classification of mental and behavioural disorders: clinical descriptions and diagnostic guidelines. *World Health Organization*, 362, 1992.

[37] Haomin Chen, Shun Miao, Daguang Xu, Gregory D Hager, and Adam P Harrison. Deep hiearchical multi-label classification applied to chest x-ray abnormality taxonomies. *Medical image analysis*, 66:101811, 2020.

[38] Ivo M Baltruschat, Hannes Nickisch, Michael Grass, Tobias Knopp, and Axel Saalbach. Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific reports*, 9(1):6381, 2019.

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[40] Bo Xiong, Michael Cochez, Mojtaba Nayyeri, and Steffen Staab. Hyperbolic embedding inference for structured multi-label prediction. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 33016–33028. Curran Associates, Inc., 2022.

# Chapter 7

# Conclusions and Future Work

This dissertation addressed the critical intersection of SOP and varying levels of supervision in remote sensing image analysis. Through a systematic progression from supervised baselines to semi-supervised hybrid frameworks and fully integrated end-to-end systems, we have developed novel methodologies that effectively leverage limited labeled data alongside abundant unlabeled samples while capturing complex label dependencies inherent in real-world visual content.

## 7.1 Summary of Contributions

The research presented in this thesis has resulted in several significant contributions to the field of SSL for SOP in image analysis, particularly in the context of remote sensing applications. These contributions address fundamental challenges in leveraging unlabeled data while preserving structured label dependencies, offering practical solutions for large-scale environmental monitoring and land cover mapping.

1. **Hybrid Semi-Supervised Framework Combining Deep Learning and Classical Methods**
   We developed a novel two-step semi-supervised learning framework that integrates CNN-based feature extraction with semi-supervised PCTs and ensemble methods. This hybrid approach leverages the representational power of deep learning architectures while maintaining the interpretability and flexibility of tree-based methods for SOP. Through extensive evaluation across multiple remote sensing datasets, we demonstrated that SSL-PCT and SSL-RForest consistently outperform their supervised counterparts across different network architectures and proportions of labeled data. The SSL-RForest method emerged as particularly effective, outperforming nine state-of-the-art self-supervised and semi-supervised deep learning methods, including BYOL, SimCLR, MoCo, and domain-specific approaches like MSMatch.

2. **End-to-End Semi-Supervised Learning Framework with Adaptive Joint Optimization**
   We introduced SSL-MAE, a unified end-to-end framework that eliminates the limitations of traditional two-stage approaches by integrating self-supervised representation learning with task-specific discriminative learning in a single optimization process. The framework employs MAE for Self-SL combined with an adaptive joint loss function that dynamically balances supervision signals throughout training. This approach consistently outperforms supervised baselines and achieves competitive performance against two-stage self-supervised methods while significantly improving computational efficiency. The adaptive joint learning mechanism demonstrates

broad applicability, successfully enhancing other self-supervised techniques, including BYOL and MoCo, when integrated into our unified framework. This contribution represents a significant advancement in eliminating the computational overhead and suboptimal feature utilization inherent in separate pre-training and fine-tuning phases.

3. **Hierarchy-Aware Semi-Supervised Learning for Complex Multi-Path Taxonomies**
   We developed HELM, the first semi-supervised image-based method capable of modeling multi-path hierarchies in HMLC. The framework integrates hierarchy-specific class tokens with graph-based structure encoders to capture complex relationships among labels, particularly in scenarios where images belong to multiple paths within a class hierarchy. HELM incorporates BYOL as a Self-SL module, enabling effective leveraging of unlabeled data while maintaining hierarchical consistency. Through evaluation on eight datasets spanning remote sensing and medical imaging domains, we demonstrated that incorporating hierarchical information significantly improves predictive performance, consistently outperforming baseline methods for both MLC and HMLC in supervised and semi-supervised settings. This work addresses a critical gap in handling real-world taxonomies that exhibit complex multi-path structures common in practical applications.

4. **Computational Efficiency Advances through Dimensionality Reduction**
   We investigated ways to improve the computational efficiency of SSL for RSI classification by applying dimensionality reduction techniques. Specifically, we employed PCA to compress high-dimensional feature spaces extracted by CNNs while preserving the advantages of semi-supervised methods. We achieved dramatic improvements in computational efficiency with 2–3 orders of magnitude speed-up in induction times while maintaining or even improving predictive performance. This contribution demonstrates that the computational benefits of SSL can be significantly enhanced without sacrificing accuracy, making these methods more practical for large-scale operational deployment.

5. **Comprehensive Evaluation and Benchmark Contributions**
   Throughout this research, we conducted extensive empirical evaluations that provide valuable insights into the design and deployment of semi-supervised methods for SOP. We established benchmarks across multiple remote sensing datasets, compared lightweight and heavy network architectures for feature extraction, and presented a comprehensive assessment of semi-supervised methods specifically tailored for multi-label classification in remote sensing contexts. Additionally, we constructed hierarchical label structures for eight datasets across remote sensing and medical imaging domains, enriching the research community with resources for future hierarchical classification research.

These contributions collectively advance the state-of-the-art in SSL for SOP, providing both theoretical insights and practical tools for addressing data scarcity challenges in complex visual recognition tasks. The developed methods offer scalable solutions that maintain or improve predictive performance while significantly reducing annotation requirements, making advanced machine learning techniques more accessible for real-world applications in environmental monitoring and land cover analysis.

## 7.2 Further Work

While this dissertation has made significant contributions to SSL for SOP, several promising directions remain for future exploration. These include advancing adaptive, hierarchy-aware modeling and visual-semantic alignment, extending the frameworks to few-shot, zero-shot, and multi-modal learning scenarios, and enhancing interpretability, efficiency, and scalability. Further developments in semantic segmentation, multispectral analysis, and cross-domain applications also represent important next steps, alongside the pursuit of stronger theoretical foundations that ensure principled and generalizable learning.

We aim to **enhance hierarchy-aware learning** through more adaptive and data-driven mechanisms. Building upon the HELM framework, advanced graph neural network architectures such as graph transformers or heterogeneous graph networks could be explored to model both hierarchical and lateral dependencies among land cover classes. Dynamic graph construction and attention-based message passing would allow models to adapt label relationships during training, improving their capacity to capture context-dependent class interactions across spatial scales. In parallel, the development of automatic hierarchy discovery methods represents an important step toward reducing reliance on predefined taxonomies such as CORINE. By learning hierarchical relationships directly from visual and semantic similarities in remote sensing data, models could uncover latent class structures that better reflect real-world landscape organization. Combining graph-based reasoning with data-driven hierarchy induction would enable next-generation semi-supervised frameworks that learn both the structure and semantics of the label space, supporting more flexible, interpretable, and scalable land cover mapping in diverse environmental contexts.

Another key direction involves **strengthening the alignment between visual features and hierarchical semantics** to enhance the effectiveness of hierarchy-aware models. One promising avenue is the development of visual-semantic initialization strategies that leverage vision-language models trained on domain-specific or hierarchically structured imagery. Such models could provide semantically meaningful and visually consistent embeddings that serve as stronger initial representations for downstream hierarchical classification tasks. Furthermore, integrating contrastive or metric-learning objectives that explicitly enforce hierarchical consistency in the embedding space may improve the model's ability to capture fine-grained relationships between related land cover classes. By unifying semantic priors from vision-language models with hierarchical visual learning, we expect to achieve better interpretability, faster convergence, and improved generalization across diverse remote sensing datasets.

We also plan to extend the proposed semi-supervised frameworks toward **few-shot and zero-shot learning** settings, where labeled samples for certain land cover types are scarce or entirely unavailable. The rich representations obtained through self-supervised pretraining, combined with the structural priors from hierarchical modeling, provide a solid foundation for addressing these challenges. Meta-learning strategies that exploit hierarchical relationships for rapid adaptation to novel or underrepresented classes could enable efficient learning in data-limited environments. Similarly, prototypical or embedding-based networks that incorporate semantic label relationships could facilitate zero-shot recognition of unseen land cover categories by transferring knowledge from related classes. These developments would substantially enhance the scalability and adaptability of remote sensing models, enabling accurate land cover mapping even in regions lacking annotated data.

As modern remote sensing increasingly relies on heterogeneous data sources such as optical, SAR, LiDAR, and hyperspectral imagery, we will extend the proposed frameworks to fully exploit **multi-modal information**. Integrating complementary spectral, spatial, and structural cues from diverse sensors can substantially improve robustness and general-

ization, particularly under challenging environmental conditions. Future work will explore modality-specific encoders within a unified semi-supervised framework, supported by cross-modal consistency objectives that encourage coherent feature representations across sensors. Furthermore, incorporating hierarchical reasoning into multi-modal fusion could help capture modality-dependent variations in class relationships, improving the interpretability of fused representations. Developing domain adaptation strategies that enable models trained on one sensor modality to transfer effectively to others would further enhance scalability and applicability in large-scale land cover mapping.

**Enhancing interpretability and transparency** is essential for the broader adoption of AI methods in remote sensing, particularly in applications that inform environmental and policy decisions. While the proposed hybrid frameworks already offer partial interpretability through tree-based components, a key objective moving forward is to develop explainability techniques that extend across all deep and hierarchical models. For deep learning modules, this may include designing attention-based visualization methods that highlight the most influential image regions for each predicted land cover class, as well as adapting gradient-based attribution approaches for multi-label and hierarchical settings. For hierarchy-aware models, visualizing how information propagates through taxonomic structures during prediction could provide deeper insights into model reasoning, consistency, and potential sources of error. Furthermore, interactive visualization tools that allow domain experts to explore and validate model outputs could bridge the gap between algorithmic transparency and real-world decision support.

Ensuring **computational efficiency and scalability** will remain a key research priority as remote sensing datasets continue to expand in resolution, frequency, and thematic diversity. Our goal is to develop lightweight and memory-efficient architectures that maintain high predictive accuracy while reducing computational costs. Model compression techniques adapted for semi-supervised and hierarchical frameworks could substantially lower overhead without compromising performance. Similarly, incremental and continual learning strategies would allow models to assimilate new unlabeled data streams as they become available, eliminating the need for full retraining. Exploring efficient attention mechanisms, adaptive token pruning, and compact hierarchical encoders would further enhance scalability, facilitating the deployment of advanced semi-supervised models in operational settings.

An important next step involves extending the proposed semi-supervised and hierarchy-aware frameworks from image-level classification to pixel-level **semantic segmentation**, enabling finer-grained land cover mapping and object delineation. Adapting SOP to segmentation will allow models to leverage spatial continuity and contextual relationships among neighboring pixels while preserving hierarchical label consistency. Incorporating **multispectral and hyperspectral** data could further enrich model representations by providing additional spectral cues that distinguish subtle material or vegetation differences not visible in RGB imagery. Hierarchical segmentation architectures that integrate spectral attention modules and class-level graph reasoning would help exploit both spectral diversity and taxonomic structure, bridging the gap between large-scale scene classification and detailed semantic understanding.

Beyond remote sensing, the developed semi-supervised and hierarchy-aware frameworks are **broadly applicable to other domains** characterized by structured and interdependent outputs. These include natural language processing tasks with hierarchical taxonomies, video understanding problems with temporal or event-based hierarchies, and audio classification tasks with taxonomic label structures. Extending these frameworks to more complex structured prediction settings such as semantic segmentation, object detection, and sequence labeling will provide valuable insights into their generality and transfer-

ability. Investigating domain-specific adaptations could further refine the models' ability to capture structured dependencies and enhance interpretability across diverse modalities.

Finally, we plan to deepen the **theoretical foundations** of the proposed methods by establishing convergence guarantees for the adaptive joint learning framework, deriving sample complexity bounds for semi-supervised hierarchical learning, and conducting generalization analyses for hybrid deep and classical machine learning approaches. A stronger theoretical understanding will provide insights for parameter selection, identify optimal training strategies, and guide the development of more principled approaches to semi-supervised structured output prediction.

The research presented in this dissertation opens numerous exciting directions for further investigation. By addressing these challenges and opportunities, future work can advance the field of semi-supervised learning for SOP, ultimately enabling more effective and efficient solutions for complex real-world visual recognition tasks while reducing the burden of manual annotation across diverse application domains.

# References


[1] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," in *IGARSS 2019-2019 IEEE international geoscience and remote sensing symposium*, IEEE, 2019, pp. 5901–5904.

[2] Q. Yuan, H. Shen, T. Li, *et al.*, "Deep learning in environmental remote sensing: Achievements and challenges," *Remote Sensing of Environment*, vol. 241, p. 111 716, 2020, ISSN: 0034-4257. DOI: https://doi.org/10.1016/j.rse.2020.111716. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425720300857.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, Oct. 2021.

[5] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[6] C. N. Silla Jr and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 31–72, 2011.

[7] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine learning*, vol. 73, no. 2, pp. 185–214, 2008.

[8] R. Copernicus, "Corine land cover," *Copernicus Land Monitoring Service. L. Monit. Serv*, 2018.

[9] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press, 2006.

[10] S. Calderon-Ramirez, S. Yang, and D. Elizondo, "Semisupervised deep learning for image classification with distribution mismatch: A survey," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 1015–1029, 2022.

[11] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NIPS*, 2017.

[12] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," *Workshop on Challenges in Representation Learning, ICML*, 2013.

[13] K. Sohn, D. Berthelot, N. Carlini, *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020, pp. 1597–1607. [Online]. Available: `https://arxiv.org/abs/2002.05709`.

[15] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [Online]. Available: `https://arxiv.org/abs/2006.07733`.

[16] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 000–16 009. [Online]. Available: `https://arxiv.org/abs/2111.06377`.

[17] R. R. Vatsavai, S. Shekhar, and T. E. Burk, "Semi-supervised classification of remote sensing images using spatial regularization," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, SIAM, 2005, pp. 362–373.

[18] A. Kili, A. Moussaoui, and S. Idbraim, "Challenges and solutions for large-scale remote sensing image annotation: A comprehensive survey," *Remote Sensing*, vol. 15, no. 4, p. 1053, 2023.

[19] N. Tsendbazar, S. de Bruin, and M. Herold, "Consistency assessment of global land cover products in the mediterranean region," *Remote Sensing*, vol. 7, no. 8, pp. 9560–9581, 2015.

[20] N. Tsendbazar, S. de Bruin, B. Mora, L. Schouten, and M. Herold, "Towards harmonized global land cover mapping: Comparing and integrating lccs-based legends," *Remote Sensing of Environment*, vol. 216, pp. 25–39, 2019.

[21] G. Weikmann, G. Perantoni, and L. Bruzzone, "A semantics-aware hierarchical self-supervised approach to classification of remote sensing images," *arXiv preprint arXiv:2510.04916*, 2025.

[22] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1144–1158, 2018.

[23] S. C. Huang *et al.*, "Self-supervised learning for medical image classification," *npj Digital Medicine*, vol. 6, no. 1, p. 70, 2023.

[24] S. Hajimiri *et al.*, "Hierarchical semi-supervised learning for medical imaging," *IEEE Transactions on Medical Imaging*, vol. 43, pp. 607–618, 2024.

[25] H. Chen, S. Miao, D. Xu, G. D. Hager, and A. P. Harrison, "Deep hiearchical multi-label classification applied to chest x-ray abnormality taxonomies," *Medical Image Analysis*, vol. 66, p. 101 811, 2020, ISSN: 1361-8415.

[26] A. Esteva, B. Kuprel, R. A. Novoa, *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[27] K. Choi, Y. Lee, C. Kim, and M. Yoon, "An effective gcn-based hierarchical multi-label classification for protein function prediction," *arXiv preprint arXiv:2112.02810*, 2021. [Online]. Available: `https://arxiv.org/abs/2112.02810`.

[28] J. Hirsch, G. Nicola, G. McGinty, *et al.*, "Icd-10: History and context," *American Journal of Neuroradiology*, vol. 37, no. 4, pp. 596–599, 2016.

[29] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5177–5186.

[30] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 478–16 488.

[31] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," *Proc. 15th Intl. Conf. on Machine Learning*, Dec. 2000.

[32] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Tree ensembles for predicting structured outputs," *Pattern Recognition*, vol. 46, no. 3, pp. 817–833, 2013, ISSN: 0031-3203.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[34] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[36] J. Levatić, M. Ceci, D. Kocev, and S. Džeroski, "Semi-supervised predictive clustering trees for (hierarchical) multi-label classification," *International Journal of Intelligent Systems*, vol. 2024, pp. 1–21, Apr. 2024.

[37] X. Chen and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020. [Online]. Available: `https://arxiv.org/abs/2003.04297`.

[38] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, Sep. 2017.

[39] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[40] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[41] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[42] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" In *NeurIPS*, 2021.

[43] M. Caron, H. Touvron, I. Misra, *et al.*, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[44] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021.

[45] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *CVPR*, 2022.

[46] M. Dehghani *et al.*, "Scaling vision transformers to 22 billion parameters," in *ICCV*, 2023.

[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[48] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 1024–1034.

[49] J. Zhou, G. Cui, S. Hu, *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[50] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1263–1272. [Online]. Available: `https://arxiv.org/abs/1704.01212`.

[51] U. Ozbulak, H. J. Lee, B. Boga, *et al.*, "Know your self-supervised learning: A survey on image-based generative and discriminative training," *Transactions on Machine Learning Research*, 2023.

[52] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[53] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[54] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[55] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[56] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2017.

[57] D. Berthelot, N. Carlini, E. D. Cubuk, *et al.*, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[58] D. Berthelot *et al.*, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *ICLR*, 2020.

[59] K. Sohn, D. Berthelot, N. Carlini, *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[60] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[61] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *International conference on machine learning*, PMLR, 2020, pp. 3897–3906.

[62] A. Oliver *et al.*, "Realistic evaluation of deep semi-supervised learning algorithms," in *NeurIPS*, 2018.

[63] Y. Kim *et al.*, "Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning," in *NeurIPS*, 2020.

[64] P. Bachman, O. Alsharif, and D. Precup, "Pseudo-ensembles and deep boltzmann machines," in *NIPS*, 2014.

[65] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[66] B. Zhang, Y. Wang, Y. Hou, *et al.*, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *NeurIPS*, 2021.

[67] E. Arazo, D. Ortego, A. Albert, N. E. O'Connor, and K. McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *ICLR*, 2020.

[68] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017.

[69] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," in *NeurIPS*, 2019.

[70] J. Lee, S. Kim, M. Lee, and H. J. Kim, "Online continual learning on hierarchical label expansion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 19 016–19 026.

[71] Y. Wang, B. Zhang, C. Gong, K. Miao, and W. Min, "Freematch: Self-adaptive thresholding for semi-supervised learning," in *NeurIPS*, 2023.

[72] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[73] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*, Springer, 2016, pp. 69–84.

[74] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *arXiv preprint arXiv:1803.07728*, 2018.

[75] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, Springer, 2016, pp. 649–666.

[76] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," in *International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2106.08254.

[77] Z. Xie, K. He, H. Zhang, *et al.*, "Simmim: A simple framework for masked image modeling," in *CVPR*, 2022. [Online]. Available: https://arxiv.org/abs/2111.09886.

[78] D. Chen, J. Bao, F. Dong, *et al.*, "Context autoencoder for self-supervised representation learning," in *CVPR*, 2022. [Online]. Available: https://arxiv.org/abs/2202.03026.

[79] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ACL, 2019, pp. 4171–4186.

[80] Q. Liu, M. He, Y. Kuang, L. Wu, J. Yue, and L. Fang, "A multi-level label-aware semi-supervised framework for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023. DOI: 10.1109/TGRS.2023.3297560.

[81] P. Gómez and G. Meoni, "Msmatch: Semisupervised multispectral scene classification with few labels," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 11 643–11 654, 2021.

[82] J. Kang, R. Fernández-Beltrán, Z. Ye, X. Tong, P. Ghamisi, and A. Plaza, "High-rankness regularized semi-supervised deep metric learning for remote sensing imagery," *Remote Sensing*, vol. 12, no. 16, 2020.

[83] Q. Li, Y. Chen, X. He, and L. Huang, "Co-training transformer for remote sensing image classification, segmentation, and detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–18, 2024. DOI: 10.1109/TGRS.2024.3354783.

[84] W. Huang, Y. Shi, Z. Xiong, Q. Wang, and X. X. Zhu, "Semi-supervised bidirectional alignment for remote sensing cross-domain scene classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 192–203, 2023.

[85] W. Miao, J. Geng, and W. Jiang, "Semi-supervised remote-sensing image scene classification using representation consistency siamese network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[86] X. Zhu and M. Bain, "B-cnn: Branch convolutional neural network for hierarchical classification," *arXiv preprint arXiv:1709.09890*, 2017.

[87] J. Chen, P. Wang, J. Liu, and Y. Qian, "Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4858–4867.

[88] B. Kolisnik, I. Hogan, and F. Zulkernine, "Condition-CNN: A hierarchical multi-label fashion image classification model," *Expert Systems with Applications*, vol. 182, p. 115 195, 2021.

[89] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," *Advances in neural information processing systems*, vol. 33, pp. 9662–9673, 2020.

[90] S. Zhang, R. Xu, C. Xiong, and C. Ramaiah, "Use all the labels: A hierarchical multi-label contrastive learning framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 16 660–16 669.

[91] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International conference on machine learning*, PMLR, 2018, pp. 5075–5084.

[92] R. Cerri, R. C. Barros, and A. C. de Carvalho, "Hierarchical multi-label classification using local neural networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 39–56, 2014, ISSN: 0022-0000.

[93] J. Chen and Y. Qian, "Hierarchical multilabel ship classification in remote sensing images using label relation graphs," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[94] M. Stoimchev, D. Kocev, and S. Džeroski, "Deep network architectures as feature extractors for multi-label classification of remote sensing images," *Remote Sensing*, vol. 15, no. 2, 538, 2023, ISSN: 2072-4292. DOI: 10.3390/rs15020538.

# Bibliography

## Publications Related to the Thesis

### Journal Articles

M. Stoimchev, D. Kocev, and S. Džeroski, "Deep network architectures as feature extractors for multi-label classification of remote sensing images," *Remote Sensing*, vol. 15, no. 2, 538, 2023, ISSN: 2072-4292. DOI: 10.3390/rs15020538.

M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Semi-supervised multi-label classification of land use/land cover in remote sensing images with predictive clustering trees and ensembles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024. DOI: 10.1109/TGRS.2024.3426981.

M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "SSL-MAE: Adaptive semisupervised learning framework for multilabel classification of remote sensing images using masked autoencoders," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 14 882–14 896, 2025. DOI: 10.1109/JSTARS.2025.3578196.

M. Stoimchev, B. Koloski, J. Levatić, D. Kocev, and S. Džeroski, "Fully- and semi-supervised hierarchical multi-label image classification with graph learning," *Under review for publication in Pattern Recognition*, 2025.

### Conference Papers

M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Hierarchical multi-label classification of land use/land cover in remote sensing images," in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Revised Selected Papers, Part III, Communications in Computer and Information Science, vol. 2560*, In press, Cham: Springer, 2026.

S. Stevanoska, M. Stoimchev, J. Levatić, and S. Džeroski, "Dimensionality reduction for efficient semi-supervised learning from remote sensing images," in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Revised Selected Papers, Part III, Communications in Computer and Information Science, vol. 2560*, In press, Cham: Springer, 2026.

# Biography

Marjan Stoimchev was born on 08.11.1994 in Kavadarci, North Macedonia. He finished his primary and secondary school in Negotino, North Macedonia. In 2013, he started his bachelor's studies at the Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University in Skopje, North Macedonia. He completed his undergraduate studies in 2018.

In the same year, he began his master's studies at the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. In January 2020, he completed his master's studies under the supervision of Prof. Dr. Vitomir Štruc and co-supervision of Dr. Klemen Grm, with the MSc thesis titled *"Learning to combine local and global image information for contactless palmprint recognition"*. For this work, he was awarded the Prešeren Award for outstanding Master's Thesis at the Faculty of Electrical Engineering, University of Ljubljana. The results of his master's research were published in the journal *Sensors* (Q1), highlighting the novelty and quality of the work.

In 2020, he joined the Department of Knowledge Technologies at the Jožef Stefan Institute, Ljubljana, Slovenia, as a visiting researcher, working under the supervision of Prof. Dr. Sašo Džeroski. In 2021, he enrolled in the PhD study program in Information and Communication Technologies at the Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, under the supervision of Prof. Dr. Sašo Džeroski and co-supervision of Dr. Dragi Kocev and Dr. Jurica Levatić.

His research interests lie in machine learning and deep learning for image analysis, with a focus on semi-supervised learning for structured output prediction. In particular, his work addresses multi-label classification (MLC) and hierarchical multi-label classification (HMLC) in domains such as remote sensing. He has published several research papers in established international journals and actively participated in international conferences. His research contributions combine expertise in deep learning, structured prediction, and remote sensing, aiming to develop robust and scalable methods for complex real-world image analysis problems.