NORA – NERC Open Research Archive

Natural Environment Research Council

UKCEH Our planet. Decoded.

Article (refereed) – postprint

Gu, Xiaowei; Kerim, Abdulrahman; Zhang, Ce; Han, Jungong; Atkinson, Peter M.; Shen, Qiang. 2026. **A semi-supervised self-organised prototype tree-based method for few-shot remote sensing scene classification.**

© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

This manuscript version is made available by the University of Bristol under the CC BY 4.0 license https://creativecommons.org/licenses/by/4.0/.

This version is available at https://nora.nerc.ac.uk/id/eprint/541792.

Copyright and other rights for material on this site are retained by the rights owners. Users should read the terms and conditions of use of this material at https://nora.nerc.ac.uk/policies.html#access.

This is an unedited manuscript accepted for publication, incorporating any revisions agreed during the peer review process. There may be differences between this and the publisher's version. You are advised to consult the publisher's version if you wish to cite from this article.

The definitive version was published in Knowledge-Based Systems, vol. 347, article 116350, 15 pp. https://doi.org/10.1016/j.knosys.2026.116350.

The definitive version is available at https://www.elsevier.com/.

Contact UKCEH NORA team at noraceh@ceh.ac.uk

The NERC and UKCEH trademarks and logos ('the Trademarks') are registered trademarks of NERC and UKCEH in the UK and other countries, and may not be used without the prior written consent of the Trademark owner.

A Semi-Supervised Self-Organised Prototype Tree-based Method for Few-Shot Remote Sensing Scene Classification

Xiaowei Gu[a], Abdulrahman Kerim[a], Ce Zhang[b,c], Jungong Han[d], Peter M. Atkinson[e,f,g], Qiang Shen[h]

[a] *School of Computer Science and Electronic Engineering, University of Surrey, Guildford, GU2 7XH, United Kingdom*
[b] *School of Geographical Sciences, University of Bristol, Bristol, BS8 1SS, United Kingdom*
[c] *UK Centre for Ecology & Hydrology, Lancaster, LA1 4AP, United Kingdom*
[d] *Department of Computer Science, the University of Sheffield, Sheffield, S10 2TN, United Kingdom*
[e] *Lancaster Environment Centre, Lancaster University, Lancaster, LA1 4YQ, United Kingdom*
[f] *Geography and Environment, University of Southampton, Southampton, SO17 1BJ, United Kingdom*
[g] *College of Surveying and Geo-Informatics, Tongji University, No.1239, Siping Road, Shanghai, 200092, China*
[h] *Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, United Kingdom*

Abstract

Few-shot learning has increasingly been explored in remote sensing scene classification as an effective approach to learning from limited examples. However, existing methods suffer from several limitations, including the need for well-annotated auxiliary datasets, limited generalisation and a tendency to overfit on training data. In this paper, a novel semi-supervised self-organised prototype tree-based method (S^3OPT) is proposed for few-shot remote sensing scene classification. S^3OPT progressively constructs a hierarchical prototype tree from image embeddings in a top-down, discriminatory manner across multiple levels of granularity, capturing inter-class similarities and intra-class variations to enable automated class separation. Using pretrained convolutional neural networks for feature extraction and exploiting pseudo-labelling, S^3OPT reduces the need for extensive manual labelling and enhances generalisation through self-training from unlabelled samples. Thanks to the prototype-based nature, S^3OPT offers high transparency and its reasoning is based on the mutual similarity between images, ensuring explainability in internal reasoning and decision-making. Extensive experiments on four widely used benchmark datasets demonstrate the great classification accuracy of the proposed S^3OPT under standard few-shot learning protocols. It achieved results superior to or on par with state-of-the-art methods for few-shot remote sensing scene classification, delivering up to a 15% accuracy increase without the requirement for computationally expensive training and/or fine-tuning.

Keywords: few-shot learning, prototype tree, remote sensing, scene classification, semi-supervised learning

1. Introduction

Remote sensing scene classification is an active research area that focuses on allocating semantic labels to remote sensing images based on their information content. As a critical task in Earth observation, it plays an important role in a wide range of geospatial applications, including urban planning, environmental monitoring, agricultural development, and land resource management [1, 2].

Although scene classification remains a challenging task, recent advances in deep learning have yielded encouraging results [3, 4, 5], surpassing conventional methods based on low-level and mid-level visual features. Deep learning models typically require large numbers of labelled training images and must undergo computationally expensive training processes to effectively capture the semantic content of remote sensing images for accurate scene classification [1, 6, 7]. However, in practice, labelling remote sensing images is a time-consuming and labour-intensive task that needs extensive domain-specific knowledge [8, 9, 10]. Without sufficient labelled images for training, deep learning models are prone to overfitting, which can lead to a significant decrease in accuracy [2, 7].

Few-shot learning has gained increasing attention in remote sensing scene classification as an effective approach to learning from limited examples without overfitting, inspired by the rapid learning ability of the human brain to recognise new categories with minimal exposure [11, 12, 13]. Few-shot learning aims to classify

Email addresses: xiaowei.gu@surrey.ac.uk (Xiaowei Gu), a.kerim@surrey.ac.uk (Abdulrahman Kerim), ce.zhang@bristol.ac.uk (Ce Zhang), jungonghan77@gmail.com (Jungong Han), pma@lancaster.ac.uk (Peter M. Atkinson), qqs@aber.ac.uk (Qiang Shen)

unseen images by leveraging knowledge from only a small number of labelled training examples, with one, five, or 10 labelled samples per class being commonly used in the literature [12, 14]. Such settings are particularly relevant in remote sensing scene classification, where annotating a large-scale, fine-resolution image database is costly and time-consuming, and labelled data for rare or newly emerging scene categories are often scarce. Few-shot learning, therefore, offers a promising, deployable approach to mitigating the reliance on extensive manual annotations and increasing classification accuracy for under-represented or entirely new classes [15, 16]. In recent years, numerous few-shot learning methods have been proposed to address this challenging task, with meta-learning-based methods emerging as the predominant paradigm in this domain [12, 17].

Despite the success recently made, few-shot learning for remote sensing images remains constrained by multiple challenges and limitations. First, remote sensing images exhibit substantial complexity, diversity and heterogeneity, characterised by high intra-class similarity, low inter-class variation, and complex geometrical structures [8, 9]. Second, the limited availability of labelled images constrains the capability of classification models to capture the discriminative characteristics of different land-use categories. Insufficient labelled training data inevitably reduces the generalisation ability of classification models [16, 18]. Another key limitation associated with meta-learning-based methods lies in the reliance on auxiliary datasets for generating simulated meta-training and meta-validation tasks [17]. Constructing well-annotated auxiliary datasets remains challenging, and their distributions may not reflect real-world conditions. Typical transfer learning-based methods, on the other hand, are computationally expensive, and their performances rely heavily on the representation learning capabilities of the employed deep learning models [19]. Finally, the deep learning models underlying most few-shot learning methods lack transparency and explainability, restricting their use in high-stakes applications [20].

To overcome the shortcomings of previous works, this paper presents a novel semi-supervised self-organised prototype tree-based (S^3OPT) method for few-shot remote sensing scene classification. The proposed S^3OPT is a prototype-based classifier with a hierarchical tree structure. The central idea of S^3OPT is to represent the few-shot remote sensing scene classification task using a hierarchy of prototypes learned from image embeddings across multiple levels of granularity, from coarse to fine. Each prototype is designated as a leaf if it comprises image embeddings of the same class, or as a branch otherwise.

The ultimate goal of S^3OPT is to progressively identify a compact set of leaf prototypes in a top-down, discriminatory manner across these granularity levels, enabling automated class separation between images of different scene categories achieved by inducing Voronoi tessellations in the image embedding space. In doing so, S^3OPT constructs a prototype tree from embeddings of a limited set of labelled remote sensing images that reveals objectively the topology of inter-class similarities and intra-class variations exhibited by these images in a natural, human-interpretable form. Importantly, the internal reasoning and decision-making of S^3OPT is based on the ensemble properties and mutual similarity of image embeddings, making the model fully explainable and trackable.

Similar to few-shot learning, semi-supervised learning aims to build classification models from limited labelled examples. While few-shot learning focuses on learning from a tiny labelled support set, semi-supervised learning additionally leverages unlabelled samples to refine the models, thereby achieving higher classification accuracy. By integrating semi-supervised learning into few-shot learning and leveraging pretrained convolutional neural networks (CNNs) without fine-tuning, S^3OPT addresses the labelling bottleneck faced by existing few-shot learning methods, particularly, these meta-learning-based ones in two complementary ways:

1. Instead of learning task-specific representations from remote sensing images, S^3OPT learns a task-specific prototype-based classifier from image embeddings extracted by pretrained CNNs [21]. This design eliminates the need for constructing auxiliary datasets by directly leveraging the strong representation capability of pretrained models and significantly reduces the computational overhead associated with CNN training or fine-tuning.
2. S^3OPT can utilise the abundant unlabelled images to refine the learned prototype tree and increase classification accuracy via self-training, after being primed with a small number of labelled images. This self-training scheme enables the classifier to continuously self-improve from unlabelled images with minimal need for human expert involvement.

To summarise, the three key innovations of the proposed S^3OPT are as follows:

1. A novel hierarchical prototype tree learned from image embeddings extracted by pretrained CNNs in a top-down, discriminatory manner, without reliance on well-annotated auxiliary datasets and providing flexibility in the choice of feature extractors.

2. A highly transparent structure objectively revealing the inter-class similarities and intra-class variations presented in images in a human-interpretable form, facilitating explainable, trackable internal reasoning and decision-making.
3. A self-training scheme that enables the prototype tree to continuously self-improve using embeddings of unlabelled images via pseudo-labelling, improving generalisation whilst minimising the need for extensive manual labelling.

A key difference between S^{3}OPT and existing prototype tree-based approaches [22, 23, 24] is that its tree structure is determined primarily by the ensemble properties and mutual distances of image embeddings in a data-driven manner, without relying on explicit splitting rules, iterative partitioning, or optimisation.

Systematic experimental investigation on widely used benchmark remote sensing datasets under standard few-shot learning settings demonstrates the efficacy of the proposed S^{3}OPT for scene classification with minimised supervision by human experts. It provides a computationally lightweight and explainable solution for few-shot remote sensing scene classification under label-scarce settings, effectively addressing the widely recognised research challenges of label scarcity, high computational costs, and model opaqueness associated with the state-of-the-art deep learning methods.

The remainder of this paper is organised as follows. A discussion of related works is given in Section 2. Section 3 describes the technical details of S^{3}OPT. Numerical examples are presented in Section 4 as the proof of concept. Section 5 concludes this paper and gives directions for future research.

2. Related Research

As S^{3}OPT is a semi-supervised method for few-shot remote sensing scene classification, the related research discussed in this section is primarily focused on remote sensing scene classification, few-shot learning and semi-supervised learning.

2.1. Remote Sensing Scene Classification

Existing methods for remote sensing scene classification can be divided into three nominal categories based on the visual features utilised, namely, low-level, mid-level and high-level [25].

Conventional methods rely primarily on low-level or mid-level visual features. Low-level methods use basic visual characteristics, such as colour, texture and structural patterns to describe the remote sensing images. Popular low-level features include colour histogram [26], scale-invariant feature transform [27] and histogram of oriented gradients [28]. However, due to the inherent complexity exhibited by remote sensing images, differentiating between scene categories based solely on low-level features is generally infeasible. Mid-level methods aim to achieve greater classification accuracy by constructing holistic representations from low-level visual features. Representative mid-level methods include spatial pyramid matching [29], bag-of-visual words [30] and locality-constrained linear coding [31]. The performance of mid-level methods depends heavily on the descriptive capabilities of the underlying low-level visual features. Both low-level and mid-level methods require careful design and are often difficult to adapt to different tasks or datasets.

High-level methods leverage semantic features learned by deep learning models for remote sensing scene classification and have dominated the field by delivering significantly higher classification accuracy than conventional methods [3]. Thanks to their strong representation learning capability, CNNs have become the primary choice, enabling end-to-end training that transforms raw data directly into classification results [32, 33, 34]. Recent studies have incorporated attention mechanisms into CNNs [35, 36] or combined CNNs with transformers [37, 38] to further enhance their ability to capture fine-grained local details in remote sensing images and establish long-range dependencies between local elements. In addition, vision transformers have recently emerged as a promising research direction for remote sensing scene classification [39, 40]. Vision-language models, which typically employ vision transformers as image encoders, have also been increasingly explored in this area [41, 42], thanks to their capability for understanding and aligning visual and textual modalities effectively.

As mentioned earlier, these deep learning models are data hungry and computationally expensive. They also have limited transparency and explainability due to their inherent “black box" nature.

2.2. Few-Shot Learning

As an important branch of modern deep learning, few-shot learning aims to classify unseen samples by learning from limited labelled training examples. Current few-shot learning methods can be divided into two major categories, namely, meta-learning-based and transfer learning-based [2].

Most existing methods for few-shot remote sensing scene classification are meta-learning-based. Meta-learning facilitates rapid adaptation of the model to novel tasks by learning from multiple different yet related

tasks, thereby empowering the model with the ability to learn quickly and effectively [34]. Meta-learning-based methods can be further divided into three subcategories, which include metric-based, optimisation-based and generation-based [6].

Metric-based methods aim to learn a similarity metric that facilitates separation between classes [43, 44, 45]. The similarity metric can take different forms, including distance measures or specialised network architectures. Well-known metric-based methods include Prototypical Networks [44] and Simple CNAPS (SCNAPS) [45]. Optimisation-based methods, such as MAML [46] and Meta SGD [47], focus on optimising parameter initialisations so that the model can adapt to novel tasks using only a few training steps [48]. Generation-based methods improve model generalisation by augmenting training data with synthetic samples created via transformations, simulations or generative models [49, 50].

The majority of meta-learning-based methods designed specifically for remote sensing scene classification are metric-based. Representative examples include, but are not limited to, CMFSL [10], HiReNet [1], ICSFF [51] and CLRL [7]. Several studies have also explored optimisation-based and generation-based methods for remote sensing scene classification [12, 52, 53]. However, constructing a well-annotated auxiliary dataset remains a key challenge for both metric-based and optimisation-based methods. Generation-based approaches are sensitive to the quality of synthetic data used for augmentation [54]. Due to the inherent complexity of remote sensing images, generating high-quality synthetic data is a highly challenging task. Furthermore, the underlying deep learning models are also subject to the limitations discussed earlier.

In contrast, transfer learning-based methods leverage knowledge acquired from other datasets through pretraining and/or fine-tuning [12]. Such methods primarily leverage contrastive learning to guide deep learning models to learn discriminative latent spaces that facilitate adaptation to differentiate new classes from limited training samples [55]. Example methods based on contrastive learning for few-shot remote sensing scene classification include, but are not limited to, MPCL-Net [56], ACLNet [34] and DBA-RMCL [57]. As contrastive learning-based methods are computationally expensive and require thoroughly designed data augmentation strategies to capture the discriminative features from images, several existing works exploit pretrained deep learning models on large-scale datasets, such as ImageNet, as image encoders. These models have demonstrated remarkable performance in standard remote sensing scene classification tasks [8, 25], and they have been applied to few-shot settings as well [12, 20]. By leveraging pretrained models without fine-tuning, transfer learning-based methods further reduce the reliance on extensive manual annotation and computational overhead. However, the performances of transfer learning-based methods rely heavily on the discriminative capabilities of the employed deep learning models for image encoding.

More recently, vision-language models, especially CLIP [58] and VisualGPT [59], have been increasingly explored for zero-shot and few-shot remote sensing scene classification [60, 61]. Pretrained using massive datasets containing hundreds of millions of text-image pairs, CLIP is capable of associating an input image with semantic descriptions. This capability facilitates open-vocabulary recognition without task-specific fine-tuning, which is particularly useful in scenarios where labelled data are scarce. The latest few-shot learning methods based on CLIP for scene classification include, but are not limited to, RemoteCLIP [62], DA-CLIP [63] and CLIP-MoA [19]. However, despite the promising performance of CLIP-based methods, key challenges remain, particularly, the significant gap between pretraining data used by CLIP and remote sensing images as well as the unique characteristics of such images, including high inter-class similarity and intra-class diversity [19, 63].

2.3. Semi-Supervised Learning

Semi-supervised learning is an important branch of machine learning that aims to build predictive models by leveraging a large number of unlabelled data together with a limited set of labelled samples [64, 65]. Semi-supervised learning reduces reliance on human supervision and offers an effective solution to tackle the labelling bottleneck, making it highly applicable to real-world scenarios where labelled training data is scarce and expensive to acquire.

Existing semi-supervised learning methods can be divided broadly into two categories, namely, transductive and inductive. Transductive methods are graph-based [66, 67]. Such methods predict the class labels of unlabelled samples by constructing a graph structure, where label information is iteratively propagated from labelled samples to unlabelled ones along the graph edges. Since transductive methods do not learn a general predictive model, they cannot make predictions on previously unseen samples. In contrast, inductive methods extend traditional supervised models by incorporating unlabelled samples during training, thereby increasing classification accuracy on unseen data [8, 68]. Classical inductive methods include wrapper methods, such as self-training [8, 68] and co-training [69], and intrinsically semi-supervised methods, such as semi-supervised

support vector machine [70].

Recently, there has been growing interest in combining semi-supervised learning with few-shot learning [71, 72]. Some semi-supervised few-shot learning methods have been proposed for remote sensing scene classification [18, 73], but this area remains under-explored.

Among existing few-shot methods, the core idea of the proposed S^{3}OPT is closest to SCNAPS [45], which is a widely used method itself and also serves as the underlying classifier of CMFSL [10]. Both S^{3}OPT and SCNAPS aim to perform few-shot learning from image embeddings by building a deterministic classifier from a limited set of labelled training images without fine-tuning the feature extractor. However, S^{3}OPT differs from SCNAPS in three key aspects:

1. S^{3}OPT identifies multiple prototypes from image embeddings across multiple levels of granularity to build a prototype tree, whereas SCNAPS uses class means as prototypes;
2. S^{3}OPT employs cosine dissimilarity as the distance measure, which is more suitable for high-dimensional embedding spaces compared with the Mahalanobis distance used in SCNAPS;
3. S^{3}OPT can self-improve using unlabelled images via pseudo-labelling after being primed with the support set, whereas SCNAPS relies solely on labelled support images to construct the classifier.

3. Proposed Approach

3.1. Overview

The overall architecture of the proposed approach is illustrated in Fig. 1. As shown, S^{3}OPT first uses a feature extractor to convert images into embedding vectors, and these embeddings then serve as the inputs for few-shot learning. The feature extractor in the proposed approach can be selected or combined from the commonly used models in computer vision tasks. In this research, CNN models pretrained on ImageNet are considered because they can capture hierarchical visual features from remote sensing images that are effective for scene classification and do not need task-specific fine-tuning. This design offers substantial flexibility in the selection of feature extractors, whilst reducing computational overhead.

The zoomed-in architecture of S^{3}OPT is also given by Fig. 1. S^{3}OPT is a prototype-based classifier with a hierarchical tree structure, where each layer corresponds to a specific granularity level from coarse to fine. Each layer is composed of multiple prototypes identified from image embeddings, with each prototype representing a distinctive local pattern at the corresponding granularity level. Prototypes can be either branches or leaves, depending on the class labels of their associated image embeddings. A prototype is categorised as a leaf if all its associated images share the same class label. Otherwise, it is a branch, and its associated images are further partitioned at the next, finer granularity level to form new prototypes. This discriminatory partitioning process repeats until all the resulting prototypes are leaves. Every leaf prototype carries a unique class label and is used to predict the class labels of unseen images during the testing stage. Branch prototypes, on the other hand, do not have class labels, but their connections, both with the branch prototypes and with leaf prototypes at the subsequent layers, jointly capture the topology of inter-class similarities and intra-class variations exhibited by the images.

In the rest of this section, the supervised learning, decision-making and self-training processes of S^{3}OPT are detailed. Note that S^{3}OPT updates its tree structure from embeddings of labelled and pseudo-labelled images in a sample-wise fashion, enabling efficient adaptation to both static and streaming application scenarios.

3.2. Supervised Learning

During supervised learning, S^{3}OPT constructs a prototype tree from embeddings of labelled training images on an image-by-image basis according to their ensemble properties and mutual distances.

Given the embedding of a training image and its corresponding class label, $< \mathbf{x}_k, y_k >$ ($y_k \in \{1, 2, ..., C\}$; C is the number of classes), S^{3}OPT firstly normalises the image embedding with its Euclidean norm ($\mathbf{x}_k \leftarrow \frac{\mathbf{x}_k}{||\mathbf{x}_k||}$), thereby transforming the Euclidean distances between the embedding and the existing prototypes into cosine dissimilarity to facilitate learning from high-dimensional problems [9]. The current learning cycle then begins and proceeds through the following four recurring steps:

Step 1. Class Mean Extraction. Given the embedding of a new training image, $\mathbf{x}_k$, the class mean of its corresponding class, $\mathbf{m}_{y_k}$ is initialised if it is the first image embedding of the class y_k presented to S^{3}OPT:

$$\mathbf{m}_{y_k} \leftarrow \mathbf{x}_k \tag{1}$$

Otherwise, the class mean $\mathbf{m}_{y_k}$ is updated as follows:

$$\mathbf{m}_{y_k} \leftarrow \mathbf{m}_{y_k} + \frac{\mathbf{x}_k - \mathbf{m}_{y_k}}{M_{y_k}};\ \mathbf{m}_{y_k} \leftarrow \frac{\mathbf{m}_{y_k}}{||\mathbf{m}_{y_k}||} \tag{2}$$

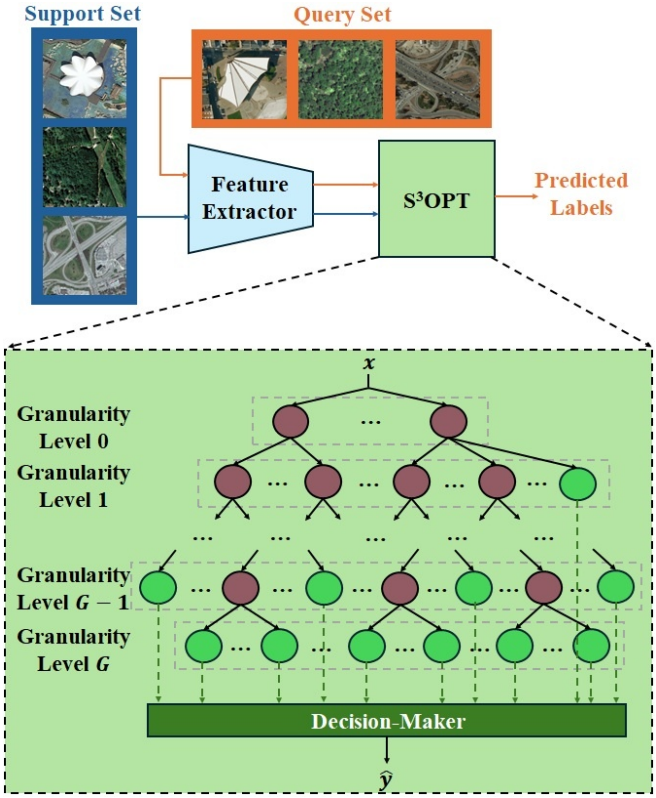


Figure 1: General architecture of S^{3}OPT (brown circles represent branch prototypes; green circles represent leaf prototypes; arrows from branch prototypes to the leaf or branch prototypes in subsequent layers indicate their affiliation relationships).

where M_i denotes the number of image embeddings of the class i that S^{3}OPT has received.

Step 2. Tree Initialisation. If $\mathbf{x}_k$ is the first image embedding received by S^{3}OPT ($k = 1$), the tree structure is initialised from scratch by creating a new leaf prototype at the first layer, corresponding to the coarsest granularity level ($j = 0$):

$$\mathbf{p}^j_{new} \leftarrow \mathbf{x}_k;\ s^j_{new} \leftarrow 1;\ l^j_{new} \leftarrow y_k \tag{3}$$

where s^j_{new} is the support (number of associated image embeddings) of the new prototype $\mathbf{p}^j_{new}$ at the j^{th} granularity level, and; l^j_{new} is its class label. $\mathbf{P}^0$ ($\mathbf{P}^0 = \{\mathbf{p}^0_{new}\}$) denotes the collection of prototypes at the first layer, corresponding to the coarsest (0^{th}) granularity level. The maximum granularity level of the tree structure is set as $G \leftarrow 0$.

Otherwise, namely, $\mathbf{x}_k$ is not the first image embedding received by S^{3}OPT ($k > 1$), the learning process proceeds directly to Step 3 for pattern matching.

Remark 1: A leaf prototype may become a branch prototype at the later learning cycles and lose its class label. In addition, the maximum granularity level G of the tree may increase in later learning cycles, allowing a finer separation of image embeddings across different classes.

Step 3. Multi-granular Pattern Matching. In this step, the most similar prototype to the training image embedding $\mathbf{x}_k$ at each layer is identified in a top-down manner using Eq. (4).

$$\mathbf{p}^j_{s^*} = \begin{cases} \arg\min_{\mathbf{p}\in\mathbf{P}^0}(||\mathbf{p}-\mathbf{x}_k||), & if\ g = 0 \\ \arg\min_{\mathbf{p}\in\mathbf{P}^{j-1}_{s^*}}(||\mathbf{p}-\mathbf{x}_k||), & else \end{cases} \tag{4}$$

where $j = 0, 1, 2, ..., G$; $||\mathbf{x}||$ denotes the Euclidean norm of $\mathbf{x}$; $\mathbf{p}^j_{s^*}$ denotes the most similar prototype at the j^{th} layer; $\mathbf{P}^{j-1}_{s^*}$ denotes the set of subordinate prototypes at the j^{th} layer that are connected to $\mathbf{p}^{j-1}_{s^*}$ at the upper layer.

One can see from Eq. (4) that the top-down search process proceeds layer by layer, comparing the embedding only to the subordinate prototypes connected to the most similar prototype at the upper layer. Hence, it

confines the comparison to a small subset of prototypes per layer, significantly reducing the search complexity compared to exhaustive comparison against all prototypes at the same layer.

The search terminates once **Condition 1** is satisfied, meaning either the end of a branch is reached or the dissimilarity between the embedding and the closest prototype at the current granularity level exceeds a predefined threshold.

$$\begin{aligned} \text{Cond. 1}: &\textit{if } (\mathbf{p}_{s^*}^j \in \mathbf{L}) \textit{ or } (||\mathbf{p}_{s^*}^j - \mathbf{x}_k|| > \gamma^j) \\ &\textit{then (the search is completed)} \end{aligned} \tag{5}$$

where $\mathbf{L}$ is the set of leaf prototypes in the tree; γ^j is the radius of influence area surrounding the prototypes at the j^{th} granularity level, serving as the similarity threshold for considering two instances as similar. γ^j is calculated as:

$$\gamma^j = \begin{cases} \sqrt{2(1-\cos(\theta_0))}, & if\ j = 0 \\ \sqrt{2(1-\cos(\frac{\theta_1}{j}))}, & else \end{cases} \tag{6}$$

Here θ_0 and θ_1 specify the maximum angle between any two similar instances at the coarsest granularity level ($j = 0$) and the first granularity level ($j = 1$), respectively. In this paper, $\theta_0 = 60^o$ and $\theta_1 = 50^o$. Note that θ_1 is a free parameter that can be chosen entirely based on user preference, without requiring prior knowledge of the problem, subject only to the constraint: $\theta_0 \geq \theta_1$.

Assuming that **Condition 1** is satisfied at the g_k^{th} granularity level ($0 \leq g_k \leq G$), there are a total of $g_k + 1$ most similar prototypes identified through the top-down search (one per granularity level), namely, $\mathbf{p}_{s^*}^0, \mathbf{p}_{s^*}^1, \cdots, \mathbf{p}_{s^*}^{g_k}$. The learning process proceeds to Step 4 for tree growth.

Remark 2: Three scenarios satisfy **Condition 1** and result in the termination of the search:

1. $\mathbf{p}_{s^*}^{g_k}$ is a leaf prototype with $l_{s^*}^{g_k} = y_k$ and is spatially close to $\mathbf{x}_k$ (e.g., $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| \leq \gamma^{g_k}$);
2. $\mathbf{p}_{s^*}^{g_k}$ is a leaf prototype with $l_{s^*}^{g_k} \neq y_k$ and is spatially close to $\mathbf{x}_k$ (e.g., $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| \leq \gamma^{g_k}$);
3. $\mathbf{x}_k$ is sufficiently distant from $\mathbf{p}_{s^*}^{g_k}$ (e.g., $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| > \gamma^{g_k}$).

Each of the three scenarios is handled differently in Step 4. Note that, in the third scenario, $\mathbf{p}_{s^*}^{g_k}$ may be either a branch or a leaf.

Step 4. Tree Growth. In this step, the tree structure expands through updating existing prototypes and adding new ones. Among the $g_k + 1$ prototypes $(\mathbf{p}_{s^*}^0, \mathbf{p}_{s^*}^1, \cdots, \mathbf{p}_{s^*}^{g_k})$ identified in Step 3, the first g_k prototypes are branches corresponding to the 0^{th} to the $g_k - 1^{th}$ granularity levels, respectively, and they are updated to move closer to $\mathbf{x}_k$ using Eq. (6).

$$\mathbf{p}_u \leftarrow \mathbf{p}_o + \frac{\mathbf{x}_k - \mathbf{p}_o}{s_o + 1};\ \mathbf{p}_u \leftarrow \frac{\mathbf{p}_u}{||\mathbf{p}_u||};\ s_u \leftarrow s_o + 1 \tag{7}$$

where $\mathbf{p}_o \in \{\mathbf{p}_{s^*}^0, \mathbf{p}_{s^*}^1, \cdots, \mathbf{p}_{s^*}^{g_k-1}\}$ denotes the original prototype; $\mathbf{p}_u$ is the updated prototype after incorporating $\mathbf{x}_k$; s_o and s_u are the supports of $\mathbf{p}_o$ and $\mathbf{p}_u$, respectively.

For the prototype $\mathbf{p}_{s^*}^{g_k}$ at the g_k^{th} granularity level, there are three possible scenarios:

Scenario 1: $\mathbf{p}_{s^*}^{g_k}$ is a leaf prototype that has the same label as $\mathbf{x}_k$ and is sufficiently similar to $\mathbf{x}_k$, namely, $l_{s^*}^{g_k} = y_k$ and $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| \leq \gamma^{g_k}$. In this case, $\mathbf{p}_{s^*}^{g_k}$ remains a leaf prototype and is updated by $\mathbf{x}_k$ using Eq. (7).

Scenario 2: $\mathbf{p}_{s^*}^{g_k}$ is a leaf prototype that has a different label from $\mathbf{x}_k$ but is sufficiently similar to $\mathbf{x}_k$, namely, $l_{s^*}^{g_k} \neq y_k$ and $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| \leq \gamma^{g_k}$. In this case, $\mathbf{p}_{s^*}^{g_k}$ is converted to a branch prototype, allowing further splitting at finer granularity levels. The minimum granularity level at which $\mathbf{p}_{s^*}^{g_k}$ and $\mathbf{x}_k$ become sufficiently dissimilar is identified by:

$$h_k = \underset{j=g_k+1, g_k+2, \ldots}{\arg\min} (||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| > \gamma^j) \tag{8}$$

Then, a new branch prototype is created at each granularity level from $g_k + 1$ to $h_k - 1$, respectively.

$$\mathbf{p}_{new}^j \leftarrow \mathbf{p}_{s^*}^{g_k};\ s_{new}^j \leftarrow s_{s^*}^{g_k} \tag{9}$$

where $j = g_k + 1, g_k + 2, ..., h_k - 1$; $\mathbf{p}_{new}^j$ is the new prototype at the j^{th} granularity level, and s_{new}^j is the corresponding support of $\mathbf{p}_{new}^j$.

At the h_k^{th} granularity level, two leaf prototypes are created:

$$\begin{cases} \mathbf{p}_{new_1}^{h_k} \leftarrow \mathbf{x}_k;\ s_{new_1}^{h_k} \leftarrow 1;\ l_{new_1}^{h_k} \leftarrow y_k \\ \mathbf{p}_{new_2}^{h_k} \leftarrow \mathbf{p}_{s^*}^{g_k};\ s_{new_2}^{h_k} \leftarrow s_{s^*}^{g_k};\ l_{new_2}^{h_k} \leftarrow l_{s^*}^{g_k} \end{cases} \tag{10}$$

Next, connections between the prototypes of the new branch are built to expand the tree structure:

$$\begin{cases} \mathbf{P}_{s^*}^{g_k} \leftarrow \{\mathbf{p}_{new}^{g_k+1}\} \\ \mathbf{P}_{new}^{j} \leftarrow \{\mathbf{p}_{new}^{j+1}\}, \quad \forall j = g_k+1, ..., h_k-2 \\ \mathbf{P}_{new}^{h_k-1} \leftarrow \{\mathbf{p}_{new_1}^{h_k}, \mathbf{p}_{new_2}^{h_k}\} \end{cases} \tag{11}$$

Finally, $\mathbf{p}_{s^*}^{g_k}$ and the newly added branch prototypes, $\mathbf{p}_{new}^{g_k+1}, \mathbf{p}_{new}^{g_k+2}, ..., \mathbf{p}_{new}^{h_k-1}$ are updated by $\mathbf{x}_k$ using Eq. (7). Furthermore, if the maximum granularity level after tree growth exceeds the original value, that is, $h_k > G$, the maximum granularity level is updated accordingly as: $G \leftarrow h_k$.

Scenario 3: $\mathbf{p}_{s^*}^{g_k}$ is sufficiently dissimilar to $\mathbf{x}_k$, namely, $||\mathbf{p}_{s^*}^{g_k} - \mathbf{x}_k|| > \gamma^{g_k}$. In this case, a new leaf prototype is created at the g_k^{th} granularity level using Eq. (3), denoted as $\mathbf{p}_{new}^{g_k}$. The new leaf prototype is connected to the tree via:

$$\begin{cases} \mathbf{P}^0 \leftarrow \mathbf{P}^0 \cup \{\mathbf{p}_{new}^{g_k}\}, \quad if \ g_k = 0 \\ \mathbf{P}_{s^*}^{g_k-1} \leftarrow \mathbf{P}_{s^*}^{g_k-1} \cup \{\mathbf{p}_{new}^{g_k}\}, \quad else \end{cases} \tag{12}$$

After the tree is updated, the current learning cycle completes, and S^{3}OPT is ready to process the embedding of the next training sample. The learning process is finished once all training samples have been processed. The prototype tree growth process is summarised in **Algorithm 1** in the form of pseudo-code.

Remark 3: Tree growth is guided by **Condition 1** based on the underlying structure and mutual distances of the image embeddings. The maximum granularity level, G (depth) of the tree is determined based on the inter-class similarity, unless explicitly constrained by the user. The greater the inter-class similarity exhibited by the image embeddings, the deeper the prototype tree tends to grow. The number of leaf prototypes at each layer is driven by intra-class diversity. The higher the intra-class diversity demonstrated by the image embeddings, the more leaf prototypes are created and the wider the tree tends to grow.

3.3. Decision-Making

After S^{3}OPT is primed with the embeddings of labelled training images using Algorithm 1, it is ready to make predictions on the class labels of unseen images. For each unlabelled image, S^{3}OPT compares its embedding, $\boldsymbol{x}_k$, with all leaf prototypes within the tree to identify the most similar leaf prototype for each class ($c = 1, 2, ..., C$):

$$\boldsymbol{\varkappa}_{c,k} = \arg\min_{\mathbf{p} \in \mathbf{L}_c}(||\mathbf{p} - \mathbf{x}_k||) \tag{13}$$

where $\mathbf{L}_c \subset \mathbf{L}$ denotes the set of leaf prototypes belonging to the c^{th} class.

The confidence score of the c^{th} class is then calculated as follows:

$$\lambda_{c,k} = \frac{\mu_{c,k} \cdot e^{-||\boldsymbol{\varkappa}_{c,k} - \mathbf{x}_k||^2} + M_c \cdot e^{-||\mathbf{m}_c - \mathbf{x}_k||^2}}{\mu_{c,k} + M_c} \tag{14}$$

where $\mu_{c,k}$ is the corresponding support of $\boldsymbol{\varkappa}_{c,k}$.

Remark 4: The confidence score, $\lambda_{c,k}$ computed by Eq. (14) integrates two complementary parts. The first part captures the similarity between $\mathbf{x}_k$ and the most similar local pattern of the c^{th} class represented by the leaf prototype $\boldsymbol{\varkappa}_{c,k}$. The second part captures the similarity between $\mathbf{x}_k$ and the global pattern of the c^{th} class represented by $\mathbf{m}_c$, akin to the approach in SCNAPS [45]. By combining both using weighted addition with the respective weights $\mu_{c,k}$ and M_c, S^{3}OPT considers both local and global data patterns in decision-making. Meanwhile, it avoids excessive reliance on local patterns, effectively reducing the risk of misleading predictions caused by class overlap and overfitting.

The class label of the image with embedding $\mathbf{x}_k$ is determined following the "winner-takes-all" strategy:

$$\hat{y}_k = \arg\max_{c=1,2,...,C}(\lambda_{c,k}) \tag{15}$$

3.4. Self-Training

During self-training, S^{3}OPT selects its high-confidence predictions on the unlabelled images to form a pseudo-labelled training set for further expansion of the prototype tree. Notably, although self-training can be performed on an image-by-image basis, processing all unlabelled images in one go (or in multiple large chunks) is recommended to maximise the information extracted from unlabelled images.

Algorithm 1 Prototype Tree Growth.

$k \leftarrow 1$
while ($< \mathbf{x}_k, y_k >$ is available) **do**
  initialise/update $\mathbf{m}_{y_k}$ by (1)/(2)
  if ($k = 1$) **then**
    initialise $\mathbf{p}^0_{new}$ by (3)
    $G \leftarrow 0$
  else
    $j \leftarrow 0$
    identify $\mathbf{p}^j_{s^*}$ by (4)
    while (**Condition 1** is not met) **do**
      $j \leftarrow j + 1$
      identify $\mathbf{p}^j_{s^*}$ by (4)
    end while
    $g_k \leftarrow j$
    for $j = 0$ to $g_k - 1$ **do**
      update $\mathbf{p}^j_{s^*}$ by (7)
    end for
    if ($||\mathbf{p}^{g_k}_{s^*} - \mathbf{x}_k|| \leq \gamma^{g_k}$) **then**
      if ($l^{g_k}_{s^*} = y_k$) **then**
        update $\mathbf{p}^{g_k}_{s^*}$ by (7) \\ *Scenario 1*
      else
        identify h_k by (8) \\ *Scenario 2*
        for $j = g_k + 1$ to $h_k - 1$ **do**
          initialise $\mathbf{p}^j_{new}$ by (9)
        end for
        initialise $\mathbf{p}^{h_k}_{new_1}$ and $\mathbf{p}^{h_k}_{new_2}$ by (10)
        establish connections by (11)
        update $\mathbf{p}^{g_k}_{s^*}$ by (7)
        for $j = g_k + 1$ to $h_k - 1$ **do**
          update $\mathbf{p}^j_{new}$ by (7)
        end for
        if ($h_k > G$) **then**
          $G \leftarrow h_k$
        end if
      end if
    else
      initialise $\mathbf{p}^{g_k}_{new}$ by (3) \\ *Scenario 3*
      establish connection by (12)
    end if
  end if
  $k \leftarrow k + 1$
end while

Given a set of embeddings of unlabelled images $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ (N denotes the cardinality of $\mathbf{U}$), S^{3}OPT firstly predicts their class labels using Eqs. (13)-(15). The predicted labels are denoted as: $\hat{\mathbf{Y}} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_N\}$.

Next, **Condition 2** is used to identify the subset of highly confident predictions to build the pseudo-labelled training set:

$$\begin{aligned} &\text{Cond. 2}: if\ (\lambda_{1^*,i} > \kappa_o \lambda_{2^*,i}) \\ &\qquad then\ (\mathbf{Z} \leftarrow \mathbf{Z} \cup \{\mathbf{u}_i\})\ and\ (\mathbf{Q} \leftarrow \mathbf{Q} \cup \{\hat{y}_i\}) \end{aligned} \tag{16}$$

where $i = 1, 2, ..., N$; $\mathbf{Z}$ is the selected subset of $\mathbf{U}$ with highly confident predicted labels; $\mathbf{Q}$ is the set of corresponding pseudo-labels; $\lambda_{1^*,i}$ and $\lambda_{2^*,i}$ are the highest and second highest confidence scores produced by S^{3}OPT on $\mathbf{u}_i$; κ_o is another free parameter that can be selected without requiring prior knowledge of the problem, subject only to the condition: $\kappa_o > 1$. In this paper, $\kappa_o = 1.1$.

Remark 5: **Condition 2** is based on the smoothness assumption. It identifies unlabelled images that exhibit greater similarity to images of a particular class than to other classes. These images carry valuable information that can be exploited to expand the prototype tree and refine the decision boundaries.

Then, the pseudo-labelled training set $\mathbf{Z}$ is used to update the prototype tree by following the same procedure as in **Algorithm 1**, where the pseudo-labels $\mathbf{Q}$ are treated as the true class labels. After the tree expansion, $\mathbf{Z}$ is removed from $\mathbf{U}$: $\mathbf{U} \leftarrow \mathbf{U} \backslash \mathbf{Z}$.

Subsequently, S^{3}OPT continues to make predictions on the remaining samples in $\mathbf{U}$, applying **Condition 2** to select additional pseudo-labelled images for self-expanding the prototype tree. The self-training process terminates once no further images with highly confident predicted class labels can be selected from $\mathbf{U}$ or $\mathbf{U}$ becomes empty.

The prototype tree growth process via self-training is summarised by **Algorithm 2**.

Algorithm 2 Prototype Tree Growth via Self-Training.

```
while (U ≠ ∅) do
  Z ← ∅
  Q ← ∅
  N ← |U|
  for i = 1 to N do
    predict ŷ_i on u_i by (13) and (15)
    if (Condition 2 is met) then
      Z ← Z ∪ {u_i}
      Q ← Q ∪ {ŷ_i}
    end if
  end for
  if (Z = ∅) then
    break
  end if
  update prototype tree with Z and Q with Algorithm 1
  U ← U\Z
end while
```

Remark 6: S^{3}OPT is less sensitive to pseudo-labelling errors thanks to the unique decision-making scheme, described in Section 3.3. By incorporating both local prototype-level and global class-level information, it prevents minor patterns formed by mislabelled samples from dominating the classification of unlabelled images.

3.5. Computational Complexity Analysis

The computational complexity of S^{3}OPT is analysed as follows.

3.5.1. Supervised Learning

During supervised learning, S^{3}OPT learns from the embeddings of labelled/pseudo-labelled training images on an image-by-image basis. Given the embedding of a particular training image, denoted as $\mathbf{x}_k$, a new learning cycle begins with Step 1. The computational complexity of updating the mean of the corresponding class is $O(W)$ (W denotes the dimensionality of $\mathbf{x}_k$). Step 2 is triggered once for initialisation only, hence, its computational complexity is negligible compared to the other steps. In Step 3, the most similar prototype to

the image embedding is identified at each layer, and the computational complexity of this process is $O(W(P_k^0 + \sum_{j=1}^{g_k} P_{s^*,k}^{j-1}))$, where P_k^0 and $P_{s^*,k}^{j-1}, \forall j$ are the respective cardinalities of $\mathbf{P}^0$ and $\mathbf{P}_{s^*}^{j-1}$ at the time instance $\mathbf{x}_k$ is observed. The tree structure expansion in Step 4 involves updating the existing prototypes and adding new prototypes. The complexity of updating existing prototypes is $O(g_k W)$, and that of adding new prototypes is negligible. Therefore, the computational complexity of a supervised learning cycle of S^{3}OPT is $O(W(P_k^0 + \sum_{j=1}^{g_k} P_{s^*,k}^{j-1}))$, which is primarily determined by Step 3. Accordingly, the overall complexity for S^{3}OPT to learn from M training samples is $O(W \sum_{k=1}^{M}(P_k^0 + \sum_{j=1}^{g_k} P_{s^*,k}^{j-1}))$.

From the above analysis, it can be observed that the computational complexity of S^{3}OPT is significantly reduced by restricting the search at each layer to the subordinate prototypes associated with the most similar prototype in the upper layer, rather than considering all prototypes in that layer.

3.5.2. Decision-Making

During decision-making, given the embedding of an unlabelled testing image, the computational complexity of identifying the most similar leaf prototype for each class is $O(WL)$, where L is the cardinality of $\mathbf{L}$. The complexity of computing the C confidence scores is $O(2WC)$. Hence, the overall computational complexity for determining the class label of an unlabelled image is $O(W(L + 2C))$.

3.5.3. Self-Training

During self-training, S^{3}OPT learns from the embeddings of a set of unlabelled training images by exploiting pseudo-labelling in an iterative manner. At the t^{th} iteration, the computational complexity of predicting the class labels of the embeddings $\mathbf{U}$ is $O(W(L_t+2C)N_t)$. Here L_t and N_t denote the cardinalities of $\mathbf{L}$ and $\mathbf{U}$ at the t^{th} iteration. The complexity of updating S^{3}OPT with the pseudo-labelled training set $\mathbf{Z}$ is $O(W \sum_{k=1}^{Z_t}(P_k^0 + \sum_{j=1}^{g_k} P_{s^*,k}^{j-1}))$ (Z_t denotes the cardinality of $\mathbf{Z}$ at the t^{th} iteration). Assuming the self-training is completed after T iterations, the overall computational cost incurred is $O(W \sum_{t=1}^{T}((L_t + 2C)N_t + \sum_{k=1}^{Z_t}(P_k^0 + \sum_{j=1}^{g_k} P_{s^*,k}^{j-1})))$.

4. Experimental Investigation

In this section, numerical experiments were conducted on publicly available datasets to evaluate the effectiveness of the proposed algorithms. The implementations were developed in Python, and all experiments were performed on a workstation equipped with an Intel Core i7–12700H processor, 64 GB RAM, and an NVIDIA RTX 3050 Ti GPU.

4.1. Configuration

4.1.1. Data Description

The following four benchmark datasets for remote sensing scene classification were used for experimental investigation.

1. **WHU-RS19** (WHU) [74] contains 19 scene categories and each category has at least 50 images.
2. **UCMerced** (UCM) [75] contains 21 scene categories with 100 images in each category.
3. **AID** [25] comprises 10,000 remote sensing images, covering 30 scene categories. The number of images per category varies from 220 to 420.
4. **NWPU-RESISC45** (NWPU) [76] contains 45 scene categories with 700 images in each category.

Images in the AID and WHU datasets have a spatial resolution of 600×600 pixels, and images in the NWPU and UCM datasets have a spatial resolution of 256×256 pixels.

Standard evaluation protocols in few-shot learning [77, 78] divide each dataset into training, validation, and testing classes. Images from the training and validation classes are employed typically for meta-training, whilst performance evaluation is carried out on the testing classes. The testing classes of the benchmark datasets are tabulated in Table 1. However, the proposed approach leverages pretrained CNN models for feature extraction and eliminates the need for fine-tuning. To ensure comparability with existing methods, performance evaluation was, therefore, conducted exclusively on the widely adopted testing classes of the benchmark datasets. This design ensures a fair and standardised comparison with state-of-the-art few-shot learning methods. Note that, for comparison with vision-language models, the same experimental protocol used by [19] is followed, where all the classes are used for accuracy evaluation.

4.1.2. Implementation Details

The proposed approach employs the following four cutting-edge CNN models pretrained on the ImageNet-1K dataset as backbones for feature extraction, which include: 1) ConvNeXtSmall; 2) ConvNeXtBase; 3)

Table 1 Selected Testing Classes of Benchmark Datasets for Performance Evaluation in Few-Shot Learning settings.

Dataset	Testing Classes
WHU	Commerical; Meadow; Pond; River; Viaduct
UCM	Beach; Golf Course; Mobile Home Park; River Sparse Residential; Tennis Court
AID	Center; Church; Forest; Industrial; River; School; Sparse Residential; Square; Storage Tanks; Viaduct
NWPU	Airport; Basketball Court; Circular Farmland; Dense Residential; Forest; Ground Track Field; Intersection; Medium Residential; Parking Lot; River

ConvNeXtLarge, and 4) ConvNeXtXLarge [21]. These models are among the best performing models on the ImageNet validation dataset, offering great accuracy, scalability and robustness across all major benchmarks. In running the experiments, the final prediction layers of the four CNN models were removed, and the activations from their last 2D average pooling layers were used as image embeddings. To ensure consistency in feature extraction, all images were resized to 248×248 pixels. From each resized image, five 224×224 segments were generated by cropping the centre and four corners from the image. The five segments were further horizontally flipped to obtain five additional segments, resulting in a total of 10 segments per image [9]. Each segment was represented by four feature vectors extracted from the four CNN models. In particular, ConvNeXtSmall produced a 768×1 dimensional feature vector, ConvNeXtBase produced a 1024×1 dimensional feature vector, ConvNeXtLarge produced a 1536×1 dimensional feature vector, and ConvNeXtXLarge produced a 2048×1 dimensional feature vector. The feature vectors of the 10 segments were first averaged and then concatenated, resulting in a 5376×1 dimensional representation with enhanced descriptive capability.

S^{3}OPT self-organises the prototype tree from image embeddings in a fully data-driven manner. The parameter θ_1 is required to determine the similarity thresholds at different granularity levels, and κ_o is used for pseudo-labelling. As mentioned earlier, default values of $\theta_1 = 50^o$ and $\kappa_o = 1.1$ are considered unless specifically declared otherwise. It will be demonstrated in the numerical examples presented later that S^{3}OPT achieves high classification accuracy across the four benchmark datasets under different experimental protocols with the same parameter setting, surpassing or at least on par with the state-of-the-art few-shot learning methods. Note that this setting merely represents a feasible choice for users' consideration. In practice, the optimal parameter values vary from problem to problem and are highly dependent on the nature of the data.

4.1.3. Evaluation Protocol

For a fair comparison against few-shot learning methods that are meta-learning-based and transfer learning-based, the standard evaluation protocol in few-shot learning was adopted [77, 78]. In particular, two experimental settings were considered: (1) five-way one-shot and (2) five-way five-shot, each with 15 query samples per class. For each dataset per experimental setting, 2000 repeated experiments were carried out in each setting by randomly sampling the testing classes. The reported accuracy was calculated as the average of 2000 repeated experiments, with a 95% confidence interval. Note that, although S^{3}OPT is applicable to both inductive and transductive inference, experiments in this research were conducted under the standard transductive setting to ensure fair comparison, unless specifically declared otherwise.

Furthermore, to compare the performance of S^{3}OPT against vision-language model-based methods, the same experimental protocol used by [19] was followed. Specifically, experiments were conducted using one, two, four, eight and 16 shots, and all the remaining images of each class were used as query images. The reported accuracy was calculated as the average of five repeated experiments to allow a certain degree of randomness.

4.2. Experimental Results and Comparisons under Standard Evaluation Protocol

The classification results of S^{3}OPT on the WHU, UCM, AID and NWPU datasets under the two few-shot learning settings are reported in Table 2. To demonstrate the efficacy of S^{3}OPT, its average classification accuracy ($\pm$95% confidence interval) is compared with 18 state-of-the-art few-shot learning methods proposed in 2023 and thereafter, thereby ensuring timely evaluation. The comparative methods include: (1) CDML [54]; (2) GDPNet [79]; (3) PA-SRM [78]; (4) CLRL [7]; (5) DiffPR-Net [2]; (6) DBA-RMCL [57]; (7) LDRNet [80];

(8) TA-MSA [17]; (9) FEL [16]; (10) HiReNet [1]; (11) ACLNet [34]; (12) PMPFSL [81]; (13) ICSFF [51]; (14) ODS-DC [53]; (15) MES^{2}L-Net [82]; (16) MPCL-Net [56]; (17) TDNet [6], and; (18) TEAW [83]. Their results on the four benchmark datasets were taken directly from the respective papers and also reported in Table 2. The best results are shown in bold for visual clarity. In addition, the results obtained by SCNAPS [45] under the exact same experimental settings used by S^{3}OPT are also provided in Table 2 as baseline. The average execution times of S^{3}OPT and SCNAPS (excluding feature extraction) on the four datasets are reported in Table 3.

One can see from Table 2 that S^{3}OPT achieved the highest classification accuracy on WHU, UCM and AID, outperforming the outstanding peers under both five-way one-shot and five-way five-shot settings. In particular, under the five-way one-shot setting, its accuracy exceeded that of the runner-up MES^{2}L-Net [82] by 4.91% on WHU, DiffPR-Net [2] by 3.84% on UCM, and GDPNet [79] by 13.70% on AID. Similarly, under the five-way five-shot setting, it surpassed the respective runner-ups, ICSFF [51], TA-MSA [17], and DBA-RMCL [57] by 2.34%, 5.50%, and 4.90% on WHU, UCM, and AID, respectively.

On the other hand, S^{3}OPT was outperformed on NWPU under both settings by three competitors, e.g., CLRL [7], DiffPR-Net [2] and MES^{2}L-Net [82]. This performance gap can be attributed to the fact that S^{3}OPT employed standard pretrained CNN models as feature extractors, whose descriptive capabilities are limited in distinguishing between images of different land-use categories. In contrast, the competitors fine-tuned their feature extractors on each dataset, leading to greater classification accuracy on NWPU because of its larger size compared to the other three datasets.

Notably, the classification accuracy of S^{3}OPT is at least 9.50% higher than that of SCNAPS, which takes the same image embeddings as inputs, on UCM, AID and NWPU under the five-way one-shot setting and at least 1.30% higher under the five-way one-shot setting. It also achieved 8.18% and 0.58% higher classification accuracy than SCNAPS on WHU under the two settings, respectively. This accuracy increase is thanks to the multi-granular prototype identification and self-training schemes of the proposed S^{3}OPT, which enable a finer separation of images of different classes in the embedding space.

A key advantage of S^{3}OPT, as noted earlier, is its flexibility in selecting feature extractors. By fine-tuning the employed CNN models or employing more powerful alternatives, the accuracy of S^{3}OPT can be further increased. It will be shown in the numerical example presented in Table 6 (see Section IV.D), by employing autoencoders to compress the extracted representations into a more compact and descriptive form, the accuracy of S^{3}OPT can be greatly increased without using any label information.

The efficiency of S^{3}OPT is another aspect worth noting. As shown in Table 3, on average, it requires less than 0.1 second to self-organise a prototype tree from the embeddings of the support and query sets. This is significantly faster than SCNAPS [45], which involves the computation of high-dimensional covariance matrices. Its computational efficiency can be further enhanced without compromising classification accuracy by employing more compact and discriminative image representations for few-shot learning.

4.3. Visual Illustration

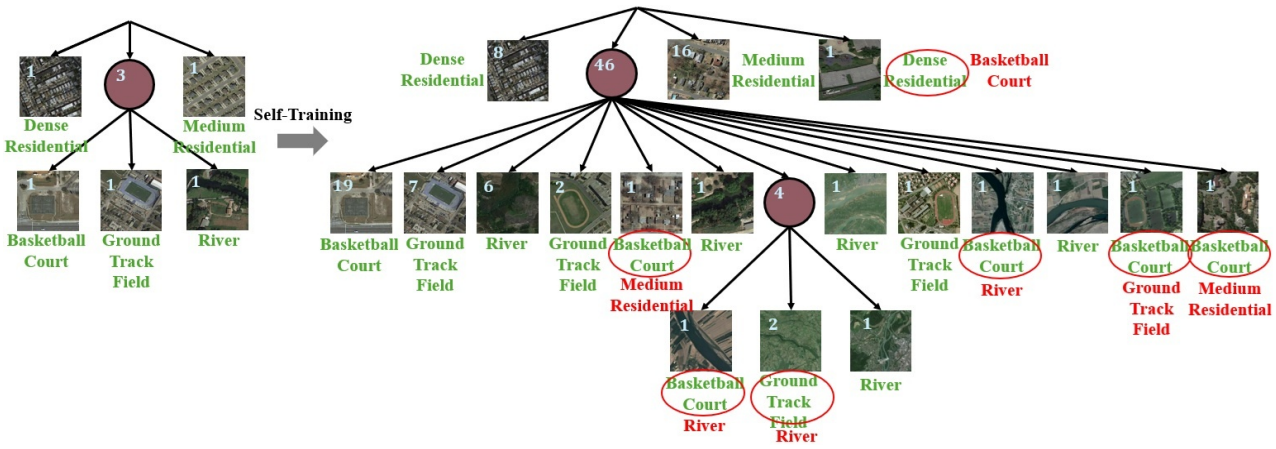


Figure 2: Visualisation of prototype tree learned from both support and query sets of NWPU dataset.

In this subsection, the prototype tree learned by S^{3}OPT from the NWPU dataset during one particular experiment under the five-way one-shot setting is visualised in Fig. 2 to provide a clearer illustration of the

Table 2 Comparison of Average Classification Accuracy (with 95% Confidence Interval) on Four Benchmark Datasets in Few-Shot Learning Setting.

Algorithm	Year	Five-Way One-Shot			
		WHU	UCM	AID	NWPU
S^3OPT	-	**97.12±0.19**	**89.20±0.35**	**87.01±0.45**	77.06±0.52
CDML [54]	2026	-	76.14±0.78	70.20±0.86	64.38±0.81
GDPNet [79]	2025	84.49±0.17	60.87±0.26	73.31±0.09	78.23±0.31
PA-SRM [78]	2025	77.49±0.71	60.79±0.28	68.75±0.36	72.65±0.43
CLRL [7]	2025	91.13±0.30	70.58±0.66	-	78.24±0.50
DiffPR-Net [2]	2025	85.43±0.21	85.36±0.24	-	81.85±0.32
DBA-RMCL [57]	2025	87.33±0.22	61.50±0.21	72.35±0.36	77.15±0.34
LDRNet [80]	2025	-	58.70±0.56	66.45±0.58	71.91±0.65
TA-MSA [17]	2025	87.24±0.32	74.20±0.49	-	68.88±0.63
FEL [16]	2024	-	55.03±0.84	61.12±0.87	64.16±0.62
HiReNet [1]	2024	-	58.60±0.80	59.43±0.66	70.43±0.90
ACLNet [34]	2024	78.30±0.32	59.74±0.46	70.86±0.31	76.13±0.20
PMPFSL [81]	2024	77.24±0.32	64.08±0.35	-	72.67±0.19
ICSFF [51]	2024	87.31±0.25	-	68.75±0.54	70.93±0.53
ODS-DC [53]	2024	-	65.93±0.94	66.28±0.89	73.93±0.90
MES^2L-Net [82]	2023	92.21±0.05	70.73±0.16	-	**86.55±0.18**
MPCL-Net [56]	2023	61.84±0.12	56.46±0.21	60.61±0.43	55.94±0.04
TDNet [6]	2023	-	-	67.48±0.51	65.85±0.53
TEAW [83]	2023	-	56.94±0.39	70.35±0.44	70.23±0.45
SCNAPS*	2020	88.94±0.32	78.03±0.43	74.41±0.50	67.54±0.50

Algorithm	Year	Five-Way Five-Shot			
		WHU	UCM	AID	NWPU
S^3OPT	-	**99.90±0.02**	**97.25±0.10**	**95.89±0.17**	89.13±0.26
CDML [54]	2026	-	90.50±0.39	85.01±0.61	84.07±0.50
GDPNet [79]	2025	91.51±0.29	78.65±0.13	88.17±0.42	86.61±0.27
PA-SRM [78]	2025	88.86±0.58	78.64±0.42	81.47±0.65	83.64±0.61
CLRL [7]	2025	97.30±0.18	87.90±0.45	-	**91.58±0.43**
DiffPR-Net [2]	2025	95.69±0.10	89.43±0.19	-	89.41±0.20
DBA-RMCL [57]	2025	93.88±0.10	78.73±0.25	90.99±0.22	86.64±0.20
LDRNet [80]	2025	-	74.18±0.39	83.77±0.36	84.50±0.40
TA-MSA [17]	2025	96.97±0.14	91.75±0.25	-	86.95±0.36
FEL [16]	2024	-	68.16±0.57	77.20±0.49	78.09±0.48
HiReNet [1]	2024	-	76.84±0.56	74.12±0.43	81.24±0.58
ACLNet [34]	2024	90.43±0.15	74.89±0.29	85.37±0.34	86.54±0.23
PMPFSL [81]	2024	88.66±0.37	81.69±0.42	-	89.54±0.17
ICSFF [51]	2024	97.56±0.08	-	87.73±0.27	86.83±0.77
ODS-DC [53]	2024	-	77.60±0.72	79.04±0.69	84.66±0.76
MES^2L-Net [82]	2023	95.03±0.04	82.92±0.12	-	91.06±0.11
MPCL-Net [56]	2023	80.34±0.54	76.57±0.07	76.78±0.08	76.24±0.12
TDNet [6]	2023	-	-	80.56±0.36	82.16±0.32
TEAW [83]	2023	-	77.50±0.27	84.62±0.25	85.57±0.25
SCNAPS*	2020	99.32±0.05	95.73±0.12	94.55±0.18	87.58±0.26

*Baseline

Table 3 Average Execution Time Comparison between S^{3}OPT and SCANPS (in Seconds)

Algorithm	Five-Way One-Shot			
	WHU	UCM	AID	NWPU
S^{3}OPT	0.0219	0.0295	0.0311	0.0323
SCANPS	8.5476	8.5687	8.5703	8.5573
Algorithm	Five-Way Five-Shot			
	WHU	UCM	AID	NWPU
S^{3}OPT	0.0260	0.0362	0.0436	0.0409
SCANPS	16.3009	16.1892	16.4432	16.2735

Table 4 Influence of θ_1 on the Performance of S^{3}OPT.

θ_1	Five-Way One-Shot		Five-Way Five-Shot	
	AID	NWPU	AID	NWPU
30^o	87.03±0.45	77.22±0.52	95.96±0.17	89.18±0.26
35^o	87.03±0.45	77.22±0.52	95.96±0.17	89.19±0.26
40^o	87.03±0.45	77.19±0.52	95.96±0.17	89.17±0.26
45^o	87.03±0.45	77.13±0.52	95.95±0.17	89.14±0.26
50^o	87.01±0.45	77.06±0.52	95.89±0.17	89.13±0.26
55^o	86.99±0.45	77.08±0.52	95.77±0.18	89.17±0.26

learning process. The prototype tree constructed from the support set is shown on the left, while the tree expanded through self-training with the query set is presented on the right. Since S^{3}OPT operates on image embeddings rather than the raw images, for clearer visualisation, each leaf prototype in Fig. 2 is illustrated using the image whose embedding exhibits the highest similarity to the prototype. Note that, in Fig. 2, brown circles represent branch prototypes; arrows between prototypes indicate their affiliation relationships; the supports of the prototypes are given in light blue, located at the upper-left corner of each node (both leaf and branch prototypes); leaf prototypes with incorrect labels are highlighted using red circles, and the correct class labels of these wrongly labelled leaf prototypes are given in red.

Fig. 2 shows that S^{3}OPT first learned a two-layer tree from the support set, consisting of one branch prototype and five leaf prototypes. Through self-training with the query set, the tree further expanded and increased its depth by acquiring an additional branch prototype and eight new leaf prototypes. Thanks to the prototype-based nature, S^{3}OPT provides a high level of transparency and explainability. The reasoning and decision-making process can be traced readily. The incorrectly labelled leaf prototypes can be identified easily within the visualised prototype tree, allowing domain experts to examine and, if necessary, manually adjust the learned prototypes and their associations.

4.4. Sensitivity Analysis

S^{3}OPT requires two free parameters to be specified by users, namely, θ_1 and κ_o. In this subsection, a sensitivity analysis was conducted to evaluate their influence on the accuracy of S^{3}OPT based on the AID and NWPU datasets under the same experimental protocols.

First, the influence of θ_1 on the performance of S^{3}OPT was evaluated. In running the experiment, the value of θ_1 varied from 30^o to 55^o with an interval of 5^o, and the value of κ_o was set as 1.1. The results of S^{3}OPT were reported in Table 4. It is shown in this table that the performance of S^{3}OPT was only marginally affected by κ_o. The accuracy remains stable on both AID and NWPU when increasing the value of κ_o from 30^o to 55^o.

Next, the influence of κ_o on the performance of S^{3}OPT was evaluated. In running the experiment, the value of κ_o varies from 1.02 to 1.22 with an interval of 0.04, and the value of θ_1 is set as 50^o. The results of S^{3}OPT are reported in Table 5. Table 5 shows that κ_o had a greater impact on the accuracy of S^{3}OPT as it directly controls pseudo-labelling via **Condition 2**. If κ_o is too small, e.g., 1.02, S^{3}OPT accumulated more pseudo-labelling errors during the self-training process, and made more mistakes during the testing. On

Table 5 Influence of κ_o on the Performance of S^{3}OPT.

κ_o	Five-Way One-Shot		Five-Way Five-Shot	
	AID	NWPU	AID	NWPU
1.02	85.79±0.40	76.47±0.49	95.61±0.17	89.13±0.25
1.06	86.68±0.42	76.86±0.51	95.97±0.16	89.22±0.26
1.10	87.01±0.45	77.06±0.52	95.89±0.17	89.13±0.26
1.14	86.68±0.48	76.83±0.53	95.64±0.18	88.92±0.26
1.18	86.05±0.51	76.37±0.54	95.31±0.19	88.40±0.26
1.22	85.16±0.53	75.74±0.55	95.01±0.20	88.58±0.26

the other hand, if κ_o is too large, e.g., 1.22, S^{3}OPT tended to be more conservative in utilising unlabelled samples to expand its prototype tree, and its accuracy also decreased because only these pseudo-labelled samples contained less new information to help build more precise classification boundaries. Based on Table 5, a suitable value range of κ_o for S^{3}OPT to achieve relatively high classification accuracy would be $[1.1, 1.14]$.

4.5. Additional Analysis

In this subsection, additional analyses were conducted to demonstrate the effectiveness of the self-training scheme in enhancing the classification accuracy of S^{3}OPT.

4.5.1. Comparison between Supervised Learning and Semi-Supervised Learning

In the first example, a supervised learning variant of S^{3}OPT, named supervised self-organised prototype tree (S^{2}OPT) was developed. S^{2}OPT followed the exact same setting as S^{3}OPT, except that it utilised only labelled images for prototype tree identification. The performance of S^{2}OPT was tested on the four benchmark datasets under the five-way one-shot and five-way five-shot settings, and the results are shown in Fig. 3. For visual comparison, the corresponding results of S^{3}OPT are included in the figure. Furthermore, the results of SCNAPS are also included in Fig. 3 as a baseline.

As shown in Fig. 3, S^{2}OPT outperformed SCNAPS under both the five-way one-shot and five-way five-shot settings in terms of classification accuracy. This accuracy improvement is attributed to the use of cosine dissimilarity as the distance measure in high-dimensional spaces and the identification of multiple prototypes to better capture the underlying data patterns. Compared with S^{2}OPT, the self-training scheme effectively enhanced the classification accuracy of S^{3}OPT by enabling the model to continue learning from unlabelled query images, thereby achieving greater classification accuracy across all datasets under both settings. A further comparison between Figs. 3a and 3b reveals that the increase in classification accuracy yielded by self-training was more significant under the five-way one-shot setting, where fewer labelled images were available to the classifiers. In addition, as shown in Fig. 4, increasing the number of query images per class from 15 to 25 consistently yielded a modest increase in the accuracy of S^{3}OPT. This increase in classification accuracy is attributed to the more information available in the larger testing set, further confirming the effectiveness of the proposed self-training scheme.

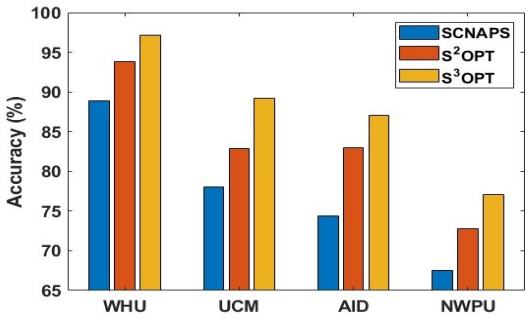
4.5.2. Performance Demonstration under Inductive Settings

Next, the efficacy of S^{3}OPT under inductive settings is investigated. Unlike the experimental protocols used before, 10 randomly selected images from each of the five testing classes are used as unlabelled training images in addition to the support and query images during the experiments. These unlabelled training images will be used by S^{3}OPT to perform self-training after being primed with the support set. The query set is then used for out-of-sample evaluation only. The results of S^{3}OPT on each dataset under the inductive experimental settings are given in Fig. 5. The results of S^{2}OPT are also included in the same figure as the baseline.

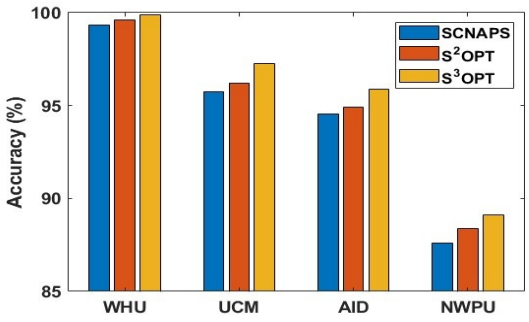
As shown in Fig. 5, the out-of-sample classification accuracy of S^{3}OPT on query images increases by learning from unlabelled training images. In addition, the accuracy increase is more pronounced when the support set is smaller, e.g., five-way one-shot. This numerical example showcases the effectiveness of S^{3}OPT under the inductive settings.

4.5.3. Ablation Analysis

The classification accuracy of S^{3}OPT with different feature extractors was evaluated on NWPU under the same experimental settings used in Table 2. Among the four CNN models used in this study, ConvNeXtXLarge

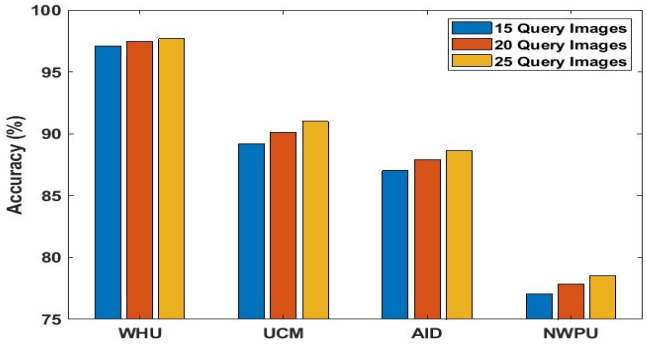


(a) Five-way one-shot.

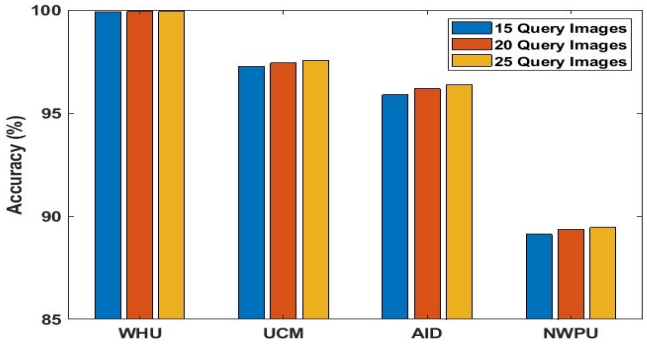


(b) Five-way five-shot.

Figure 3: Classification accuracy comparison between SCNAPS, S^2OPT and S^3OPT under transductive settings.

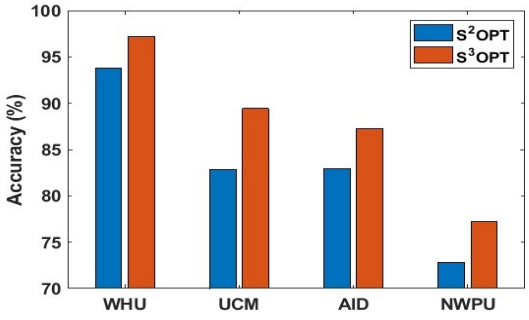


(a) Five-way one-shot.

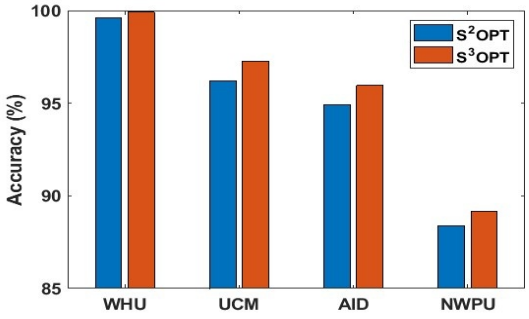


(b) Five-way five-shot.

Figure 4: Variation in classification accuracy with different numbers of query images per class.



(a) Five-way one-shot.



(b) Five-way five-shot.

Figure 5: Classification accuracy comparison between S^2OPT and S^3OPT in inductive settings.

Table 6 Ablation Analysis Results on the NWPU Dataset.

ConvNeXt				Fine	Five-Way	Five-Way
Small	Base	Large	XLarge	Tuning	One-Shot	Five-Shot
×	×	×	✓	×	76.34±0.53	88.80±0.26
×	×	✓	✓	×	77.00±0.52	89.08±0.26
×	✓	✓	✓	×	76.93±0.52	89.01±0.26
✓	✓	✓	✓	×	77.06±0.52	89.13±0.26
✓	✓	✓	✓	✓	78.72±0.56	90.62±0.24

achieved the highest Top-1 accuracy on the ImageNet validation dataset, followed by ConvNeXtLarge, while ConvNeXtSmall achieved the lowest. Based on this, three variants of S^{3}OPT were considered, each using a different combination of CNN models for feature extraction:

1. ConvNeXtBase, ConvNeXtLarge, and ConvNeXtXLarge;
2. ConvNeXtLarge and ConvNeXtXLarge;
3. ConvNeXtXLarge only.

The three variants followed the same procedure for feature extraction as given in Section 4.1.2 to ensure a fair comparison. Their results are reported in Table 6, along with the baseline results obtained using all four CNN models.

To demonstrate the potential of S^{3}OPT when combined with stronger feature extractors, self-supervised fine-tuning (SSFT) was utilised to enhance the descriptive capability of the four CNN models by performing dimensionality reduction on the extracted feature vectors, rather than fine-tuning the CNN models themselves. For each CNN model, a three-layer perceptron network with sigmoid activation was trained in a self-supervised manner on 40% of randomly sampled images from the four datasets, learning to reconstruct the original feature vectors at the output layer without using label information. The hidden layer size of the network was set to one quarter of the input dimensionality. After 100 training epochs, the output layer was discarded, yielding an encoder for dimensionality reduction. The compressed feature vectors produced by the four encoders (one per CNN model) were then concatenated into a 1344×1 dimensional representation for each image. The compressed presentations of the images were used as inputs to S^{3}OPT. The results obtained with these compressed representations under the same experimental settings are also reported in Table 6.

As shown in Table 6, the classification accuracy of S^{3}OPT increased steadily when more powerful CNN models were used for feature extraction. Moreover, its accuracy can be further increased by applying SSFT techniques to increase the descriptive capability of the feature extractors. Specifically, after SSFT, S^{3}OPT achieved the second and third highest accuracy on NWPU among the state-of-the-art few-shot learning methods tabulated in Table 2 under the five-way one-shot and five-way five-shot settings, respectively.

4.5.4. Performance Comparison against Visual-Language Model-based Methods

Finally, the few-shot learning performance of S^{3}OPT was compared against four visual-language model-based methods, including linear probe CLIP (LP-CLIP), CoOp [84], CLIP-Adaptor (CLIP-A) [85] and CLIP-MoA [19], on the four datasets, following the same evaluation protocol used in [19]. The results of S^{3}OPT and the four comparative methods are presented in Fig. 6. Note that, during the experiments, SSFT was utilised to fine tune the CNN models used for feature extraction, following the same process described in Section 4.5.3. The results of LP-CLIP, CoOp, CLIP-A and CLIP-MoA were taken directly from [19].

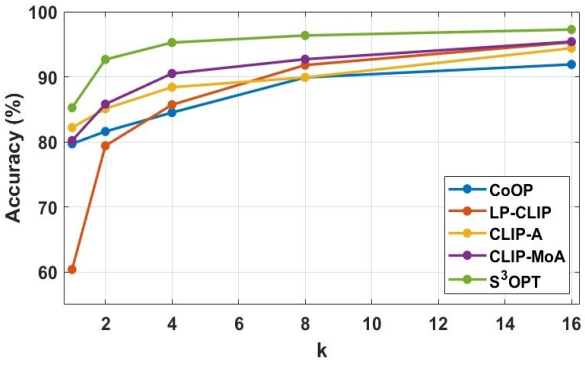
One can see from Fig. 6 that the proposed S^{3}OPT consistently outperformed the four visual-language model-based methods on WHU, UCM and AID in terms of classification accuracy. It achieved higher accuracy than the four competitors in one, two and four shots on NWPU. This example further demonstrates the effectiveness of S^{3}OPT for few-shot learning.

4.6. Discussion

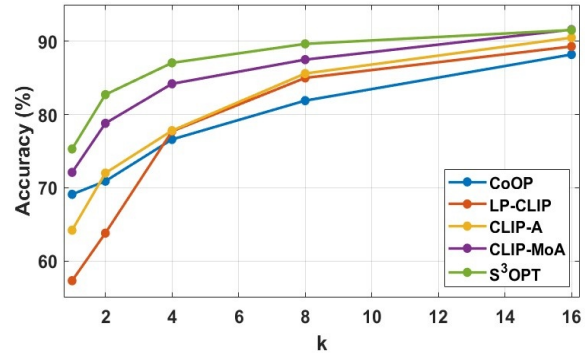
The systematic experimental investigation above shows that the proposed S^{3}OPT performs well for the benchmark remote sensing datasets considered in this research, achieving classification accuracies that surpass the state-of-the-art few-shot learning methods on the WHU, UCM and AID datasets under the commonly used few-shot experimental settings, and on par with the best-performing methods on NWPU.

S^{3}OPT is particularly well-suited for application scenarios where annotated images of certain scene categories are scarce, such as emerging or rapidly evolving remote sensing tasks. Representative scenarios include environmental monitoring, urban development mapping, agricultural crop surveillance, and disaster damage assessment, where new scene categories frequently emerge and the ability to generalise from only a few examples is of great importance. As it is designed to construct a task-specific classifier from image embeddings rather than adapting feature extractors to learn task-specific representations, S^{3}OPT can be treated as an add-on module and deployed across different tasks without introducing any changes to the existing representation learning modules. Additionally, its interpretable prototype-based tree structure makes it particularly useful in applications that require transparency and interpretability in decision-making, such as land-use policy planning or ecological research.

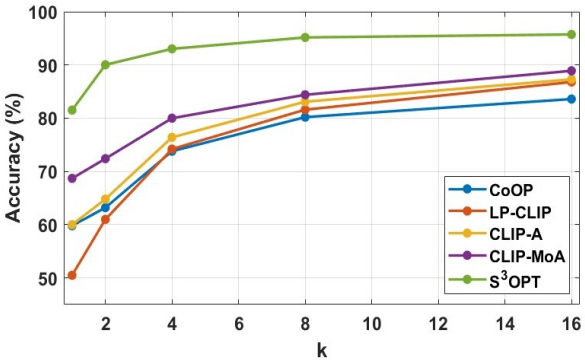
One limitation of S^{3}OPT is associated with its largely hand-crafted mechanism, which relies on soft thresholds for tree growth, prototype identification and pseudo-labelling. While this design was demonstrated here



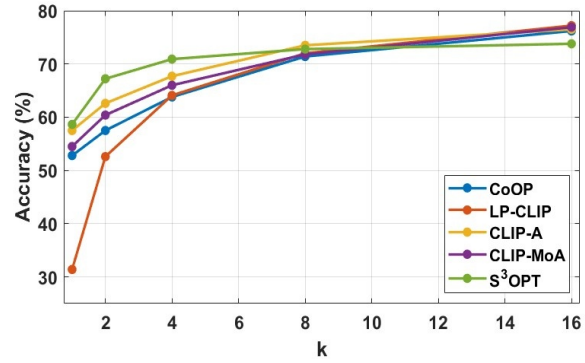
(a) WHU.



(b) UCM.



(c) AID.



(d) NWPU.

Figure 6: Classification accuracy comparison between S^3OPT and four vision-language model-based methods on four benchmark datasets.

to be highly effective on the considered benchmark remote sensing datasets under standard few-shot learning settings, it may construct an over-complex prototype tree when applied to datasets with extremely high intra-class variability and inter-class similarity. Although the risk of overfitting is effectively reduced thanks to the unique decision-making mechanism of S^{3}OPT that combines local prototype-level and global class-level information, the importance of local data patterns in decision-making is also reduced due to the smaller supports associated with leaf prototypes. This can decrease the classification accuracy of S^{3}OPT, particularly on datasets with high inter-class similarity. Incorporating adaptive thresholds that self-adjust according to data characteristics is a promising direction for future research, which could provide additional adaptability and flexibility at the cost of potentially higher computational complexity.

Another limitation of S^{3}OPT is associated with its self-training scheme via pseudo-labelling. Under the few-shot learning settings, the scarcity of labelled data makes it challenging to ensure pseudo-label quality. One approach is to increase the value of κ_o such that **Condition 2** will be stricter and only allow the most confident predictions to be used for self-training. However, there is a risk of information loss as these few images satisfying **Condition 2** may be insufficient for effectively expanding the knowledge base. On the other hand, if κ_o is set too small, **Condition 2** becomes overly permissive, causing the accumulation of pseudo-labelling errors, leading to a decrease in classification accuracy. Therefore, κ_o must be set properly to ensure **Condition 2** is neither overly permissive nor overly strict. This trade-off remains a fundamental research challenge in semi-supervised learning [65].

In practice, when applying S^{3}OPT to an unseen dataset with limited prior knowledge, the default parameter settings for θ_1 and κ_0 given in Section 4.1.2 may be adopted, particularly when only a very limited number of labelled training examples are available. Under this default configuration, S^{3}OPT has consistently demonstrated strong classification performance across the extensive numerical experiments conducted in this study. When sufficient data are available to construct a separate validation set, the two parameters may also be further tuned to maximise the validation performance of S^{3}OPT. Furthermore, the average cosine dissimilarity between image embeddings may also be considered when determining the value of θ_1. The parameter κ_0 may be adjusted to control the trade-off between aggressive and conservative prototype tree expansion, as discussed in Section 4.4.

One may also note that, the visual example (Fig. 2) is derived from the learned prototypes obtained in a representative experiment conducted under the few-shot learning setting with a limited number of images. Due to the inherent characteristics of remote sensing datasets, images often exhibit strong inter-class similarities and substantial intra-class variations, which can occasionally lead to ambiguous predictions. It is important to distinguish this aleatory uncertainty, which originates from the inherent variability in the data, from epistemic uncertainty, which arises from limitations in the model's learning process.

5. Conclusion

This paper presented a novel few-shot learning method, named S^{3}OPT, for remote sensing scene classification. S^{3}OPT self-organises a hierarchy of prototypes from both labelled and unlabelled remote sensing images in a top-down, discriminatory manner across multiple levels of granularity to capture underlying image patterns while objectively revealing inter-class similarities and intra-class variations. By employing state-of-the-art pretrained CNNs for feature extraction, S^{3}OPT leverages powerful representation capabilities without requiring computationally expensive training or fine-tuning, while minimising the reliance on extensive manual labelling. Moreover, this design ensures that S^{3}OPT is decoupled to the choice of feature extractors and offers flexibility to further increase classification accuracy through incorporating more advanced feature extraction techniques. Systematic experimental investigations demonstrate the efficacy of the proposed S^{3}OPT as a promising method for few-shot learning.

There are a few considerations for future research:

First, as mentioned earlier, it would be useful to develop a data-driven mechanism for estimating the soft thresholds directly from image embeddings, thereby providing S^{3}OPT with greater adaptability and flexibility to varying data distributions and scene complexities.

Second, it would be worthwhile to use more advanced pseudo-labelling strategies for more effective self-training. Training a more accurate teacher model could also be a feasible option to further increase the accuracy of pseudo-labelling.

Third, the CNN models pretrained on natural images and employed by S^{3}OPT for feature extraction may be insufficient to fully capture the discriminative characteristics of remote sensing images across diverse scene categories. Exploring advanced models trained on large-scale remote sensing datasets would therefore be valuable.

Fourth, fine-tuning the standard CNN models on remote sensing images in a self-supervised manner, without requiring any additional labelled data, has shown to be effective in increasing the classification accuracy of S^3OPT (Table 6). This also represents a promising direction for further exploration.

Last, one may also explore the use of image encoders from cutting-edge vision-language models for extracting representations from remote sensing images, benefiting from their extensive pretraining on large-scale image-text pairs.

Acknowledgement

This research was supported by Engineering and Physical Sciences Research Council under Grant EP/X027732/1.

References

[1] F. Tian *et al.*, "Hirenet: hierarchical-relation network for few-shot remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–10, 2024.

[2] Y. Zhu *et al.*, "Diffpr-net: few-shot remote sensing scene classification based on generative diffusion and prototype rectified model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–13, 2025.

[3] Y. Li *et al.*, "Deep learning for remote sensing image classification: a survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, 2018.

[4] L. Ma *et al.*, "Deep learning in remote sensing applications: a meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019.

[5] S. Lu *et al.*, "Vision foundation models in remote sensing: a survey," *IEEE Geoscience and Remote Sensing Magazine*, 2025.

[6] B. Wang *et al.*, "Tdnet: a novel transductive learning framework with conditional metric embedding for few-shot remote sensing image scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 4591–4606, 2023.

[7] L. Wang *et al.*, "Learning class-aware local representations for few-shot remote sensing scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 13 225–13 237, 2025.

[8] X. Gu *et al.*, "A semi-supervised deep rule-based approach for complex satellite sensor image analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2281–2292, 2020.

[9] ——, "A self-training hierarchical prototype-based ensemble framework for remote sensing scene classification," *Information Fusion*, vol. 80, pp. 179–204, 2022.

[10] B. Xi *et al.*, "Few-shot learning with class-covariance metric for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 5079–5092, 2022.

[11] M. Singha *et al.*, "Applenet: visual attention parameterized prompt learning for few-shot remote sensing image generalization using clip," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2024–2034.

[12] C. Qiu *et al.*, "Few-shot remote sensing image scene classification: recent advances, new baselines, and future trends," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 209, pp. 368–382, 2024.

[13] Y. Li *et al.*, "Hpmf: hypergraph-guided prototype mining framework for few-shot object detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

[14] J. Seo, H. Jung, and S. Lee, "Self-augmentation: generalizing deep networks to unseen classes for few-shot learning," *Neural Networks*, vol. 138, pp. 140–149, 2021.

[15] W. Wang *et al.*, "A survey of zero-shot learning: settings, methods, and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–37, 2019.

[16] Y. Rong, Q. Kong, and G. Wang, "Few-shot remote sensing scene recognition via feature enhancing learning," *IEEE Access*, 2024.

[17] X. Li *et al.*, "Ta-msa: a fine-tuning framework for few-shot remote sensing scene classification," *Remote Sensing*, vol. 17, no. 8, p. 1395, 2025.

[18] H. Su *et al.*, "Iterative semi-supervised learning with few-shot samples for coastal wetland land cover classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[19] Z. Fu, H. Yan, and K. Ding, "Clip-moa: visual-language models with mixture of adapters for multitask remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, no. 55, pp. 1–17, 2025.

[20] G. Lee *et al.*, "Unlocking the capabilities of explainable few-shot learning in remote sensing," *Artificial Intelligence Review*, vol. 57, no. 7, p. 169, 2024.

[21] Z. Liu *et al.*, "A convnet for the 2020s," in *IEEE/CVF Conference on Computer Vision and Pattern Recognitio*, 2022, pp. 11 976–11 986.

[22] P. Hase *et al.*, "Interpretable image recognition with hierarchical prototypes," in *AAAI Conference on Human Computation and Crowdsourcing*, 2019, pp. 32–40.

[23] M. Nauta, R. V. Bree, and C. Seifert, "Neural prototype trees for interpretable fine-grained image recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 933–14 943.

[24] M. Liang *et al.*, "Interpretable inference and classification of tissue types in histological colorectal cancer slides based on ensembles adaptive boosting prototype tree," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 12, pp. 6006–6017, 2023.

[25] G. Xia *et al.*, "Aid: a benchmark dataset for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.

[26] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[27] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[28] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[29] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2169–2178.

[30] J. Yang *et al.*, "Evaluating bag-of-visual-words representations in scene classification," in *International Workshop on Multimedia Information Retrieval*, 2007.

[31] J. Wang *et al.*, "Locality-constrained linear coding for image classification," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367.

[32] W. Wang, Y. Chen, and P. Ghamisi, "Transferring cnn with adaptive learning for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[33] T. Zhang *et al.*, "Dcnnet: a distributed convolutional neural network for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–18, 2023.

[34] Y. Xu *et al.*, "Attention-based contrastive learning for few-shot remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–17, 2024.

[35] X. Chen *et al.*, "Attention-aware deep feature embedding for remote sensing image scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, pp. 1171–1184, 2023.

[36] W. Dai *et al.*, "A multi-scale dense residual correlation network for remote sensing scene classification," *Scientific Reports*, vol. 14, no. 1, p. 22197, 2024.

[37] X. Tang *et al.*, "Emtcal: efficient multiscale transformer and cross-level attention learning for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[38] Y. Duan *et al.*, "Stmsf: Swin transformer with multi-scale fusion for remote sensing scene classification," *Remote Sensing*, vol. 17, no. 4, p. 668, 2025.

[39] P. Lv *et al.*, "Scvit: a spatial-channel feature preserving vision transformer for remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[40] Z. Sha and J. Li, "Mitformer: a multiinstance vision transformer for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022.

[41] X. Li *et al.*, "Vision-language models in remote sensing: current progress and future trends," *IEEE Geoscience and Remote Sensing Magazine*, vol. 12, no. 2, pp. 32–66, 2024.

[42] M. Rahhal, Y. Bazi, and M. Zuair, "Lora-clip: efficient low-rank adaptation of large clip foundation model for scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 22, no. 55, pp. 1–5, 2025.

[43] G. Koch *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015.

[44] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077—4087.

[45] P. Bateni *et al.*, "Improved few-shot visual classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 493–14 502.

[46] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017, pp. 1126–1135.

[47] Z. Li *et al.*, "Meta-sgd: learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[48] O. Vinyals *et al.*, "Matching networks for one-shot learning," in *Advances in Neural Information Processing Systems*, 2016.

[49] E. Schonfeld *et al.*, "Generalized zero- and few-shot learning via aligned variational autoencoders," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8247–8255.

[50] H. Wang *et al.*, "Towards well-generalizing meta-learning via adversarial task augmentation," *Artificial Intelligence*, vol. 317, p. 103875, 2023.

[51] C. Yang *et al.*, "Icsff: information constraint on self-supervised feature fusion for few-shot remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–12, 2024.

[52] D. Alajaji and H. Alhichri, "Few shot scene classification in remote sensing using meta-agnostic machine," in *IEEE Conference on Data Science and Machine Learning Applications*, 2020, pp. 77–80.

[53] Z. Dong, B. Lin, and F. Xie, "Optimizing few-shot remote sensing scene classification based on an improved data augmentation approach," *Remote Sensing*, vol. 16, no. 3, p. 525, 2024.

[54] W. Zhao *et al.*, "First-order cross-domain meta learning for few-shot remote sensing object classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026, doi:10.1109/TPAMI.2026.3656494.

[55] C. Liu *et al.*, "Learning a few-shot embedding model with contrastive learning," in *AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8635–8643.

[56] J. Ma *et al.*, "Multipretext-task prototypes guided dynamic contrastive learning network for few-shot remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023.

[57] R. Dong *et al.*, "Dba-rmcl: refined metric contrastive learning with dual-branch attention for few-shot remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 63, pp. 1–16, 2025.

[58] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021, pp. 8748–8763.

[59] J. Chen *et al.*, "Visualgpt: data-efficient adaptation of pretrained language models for image captioning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 030–18 040.

[60] C. Qiu *et al.*, "Few-shot remote sensing image scene classification: recent advances, new baselines, and future trends," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 209, pp. 368–382, 2024.

[61] W. Zhang *et al.*, "Earthgpt: a universal multimodal large language model for multisensor image comprehension in remote sensing domain," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–20, 2024.

[62] F. Liu *et al.*, "Remoteclip: a vision language foundation model for remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, no. 55, pp. 1–16, 2024.

[63] D. Deng and P. Yao, "Dual-alignment clip: task-specific alignment of text and visual features for few-shot remote sensing scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, no. 55, pp. 19 260–19 272, 2025.

[64] X. Yang *et al.*, "A survey on deep semi-supervised learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934–8954, 2022.

[65] J. Van Engelen and H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[66] M. Wang *et al.*, "Scalable semi-supervised learning by efficient anchor graph regularization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1864–1877, 2016.

[67] X. Fan *et al.*, "Fast semi-supervised classification based on anchor graph," *Information Sciences*, vol. 699, p. 121786, 2025.

[68] X. Gu, "A self-training hierarchical prototype-based approach for semi-supervised classification," *Information Sciences*, vol. 535, pp. 204–224, 2020.

[69] Z. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[70] O. Chapelle, V. Sindhwani, and S. Keerthi, "Optimization techniques for semi-supervised support vector machines," *Journal of Machine Learning Research*, vol. 9, no. 2, pp. 203–233, 2008.

[71] X. Li *et al.*, "Learning to self-train for semi-supervised few-shot classification," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 10 276–10 286.

[72] Z. Yu *et al.*, "Transmatch: a transfer-learning scheme for semi-supervised few-shot learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 853–12 861.

[73] H. Ji *et al.*, "Semi-supervised few-shot classification with multitask learning and iterative label correction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.

[74] G. Xia *et al.*, "Structural high-resolution satellite image indexing," in *ISPRS TC VII Symposium - 100 Years ISPRS*, vol. 38, 2010, pp. 298–303.

[75] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.

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

[77] Y. Xu *et al.*, "Attention-based contrastive learning for few-shot remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–17, 2024.

[78] Y. Jia *et al.*, "Few-shot remote sensing scene classification via parameter-free attention and region matching," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 227, pp. 265–275, 2025.

[79] N. Liu, B. Liu, J. He, and Y. Xiao, "Gdpnet: gated dynamic prototype network for few-shot remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 22, no. 55, pp. 1–5, 2025.

[80] A. Qin *et al.*, "Local-descriptors-based rectification network for few-shot remote sensing scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 9566–9581, 2025.

[81] S. Wang *et al.*, "Personalized multiparty few-shot learning for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–15, 2024.

[82] J. Li *et al.*, "Multiform ensemble self-supervised learning for few-shot remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023.

[83] K. Cheng *et al.*, "Teaw: text-aware few-shot remote sensing image scene classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023, pp. 1–5.

[84] K. Zhou *et al.*, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[85] P. Gao *et al.*, "Clip-adapter: better vision-language models with feature adapters," *International Journal of Computer Vision*, vol. 132, no. 2, pp. 581–595, 2024.