Few-Shot Open-Set Hyperspectral Pixelwise Classification With Spatial Aware Label Propagation

Matheus B. Pereira, Jefersson A. dos Santos, *Senior Member, IEEE*, and Laurent Najman, *Senior Member, IEEE*

***Abstract*—Pixelwise classification of hyperspectral images (HSI) is a fundamental task in remote sensing, but its deployment in real-world applications is often limited by the scarcity of labeled data and the presence of unknown classes at inference time. In this work, we propose Few-shot Open-set hyperspectral recognition from Self-Supervised Learning training (FOSSeL), a unified framework for few-shot, open-set hyperspectral pixelwise classification that does not require supervised training. The proposed approach first leverages self-supervised learning to construct a discriminative feature space from unlabeled data. Then, given a small set of labeled seed pixels, class predictions are obtained through spatially-aware graph-based label propagation over superpixel representations, enabling the joint exploitation of spectral similarity and spatial consistency. To address the open-set scenario, we introduce simple yet effective strategies to identify pixels that do not belong to the known classes. We evaluate FOSSeL on three benchmark hyperspectral datasets under multiple few-shot settings. The results show that the proposed framework achieves competitive or superior performance compared to fully supervised methods while maintaining robustness in the presence of unknown classes. These findings highlight the potential of combining self-supervised feature learning with graph-based inference for practical HSI analysis in low-annotation and open-world scenarios.**

***Index Terms*—Few-shot, hyperspectral, open-set, pixelwise classification, remote sensing, segmentation, self-supervised.**

I. Introduction

EARTH observation technologies have advanced quickly in recent years, both in terms of modern infrastructure for new sensors/platforms and newly developed data processing tools, which is an important topic in the research community [1]. Remote sensing data plays a crucial role in this context, allowing a comprehensive study of the Earth's surface at different scale levels [2], with applications in land use, environmental monitoring, natural resource management, among other tasks [3].

Hyperspectral imaging techniques are essential for several remote sensing applications, allowing for the discrimination of subtle spectral differences [4]. The result of such techniques is hyperspectral images (HSI), which may contain hundreds of spectral bands that exceed the perception range of the human eye [5]. In this type of imagery, each pixel can be viewed as a high-dimensional vector representing the spectral reflectance of the inspected surface at a specific wavelength [4].

One of the main problems when dealing with hyperspectral data processing is pixelwise classification, also commonly named in the literature as HSI classification or recognition. In this task, the objective is to classify each pixel in the input image into a land cover class, such as buildings, rivers, roads, and different types of vegetation, among others [5]. This is an important field of research in the hyperspectral community, as proposed methods often face challenges related to the large spatial variability of spectral signatures and limited available training samples [4].

Following a similar trend to other computer vision tasks, HSI pixelwise classification results have been successfully improved in recent years due to advances in deep learning techniques [4], [6]. Despite the existence of robust classification methods in the literature, an important problem arises when most of the proposed approaches work under a closed-set scenario in which all training and testing pixels come from the same label space [7]. In real-world applications, however, this assumption is often difficult to satisfy, since the learning samples are limited, and will hardly represent the whole universe of potential classes to appear in the deployment phase. As a result, the existence of unknown or novel classes will lead to false positives, reducing the precision of the model [6].

Open-set pixelwise classification methods aim at solving the aforementioned problem of identifying pixels that do not belong to the same set of classes seen during training, while also correctly classifying the pixels from the known classes. Although there is a relatively high number of papers in the literature tackling the open-set problem (OSR) with typical natural images (RGB), only a few approaches are focused specifically

Received 30 April 2026; revised 9 July 2026 and 10 August 2026; accepted 2 September 2026. Date of publication 9 September 2026; date of current version 29 September 2026. This work was supported in part by the Serrapilheira Institute under Grant #R-2011-37776, in part by the Brazil National Council for Scientific and Technological Development (CNPq), in part by the International Cooperation Program CAPES/COFECUB Fondation, Université Gustave Eiffel, and in part by the CAPES – Brazilian Federal Agency for Support and Evaluation of Graduate Education within the Ministry of Education of Brazil. *(Corresponding author: Matheus B. Pereira.)*

Matheus B. Pereira is with the Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG 31270-010, Brazil, and also with the Laboratoire d'Informatique Gaspard Monge, Université Gustave Eiffel, 93162 Noisy-le-Grand, France (e-mail: matheuspereira@dcc.ufmg.br).

Jefersson A. dos Santos is with the Department of Computer Science, University of Sheffield, Sheffield S102TN, , U.K. (e-mail: j.santos@sheffield.ac.uk).

Laurent Najman is with the Mathematics department at Khalifa University, Abu Dhabi, UAE, on leave from the Laboratoire d'Informatique Gaspard Monge, ESIEE Paris, Université Gustave Eiffel, 93162 Noisy-le-Grand, France (e-mail: laurentalain.najman@ku.ac.ae).

Digital Object Identifier 10.1109/JSTARS.2026.3732597

© 2026 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

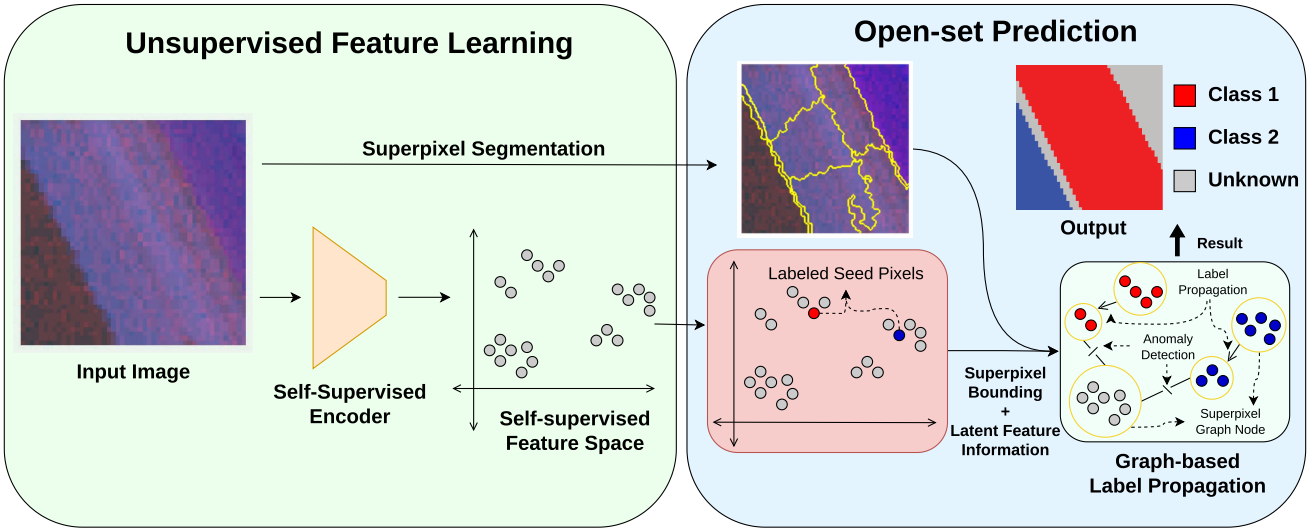


Fig. 1. Overview of the proposed FOSSeL framework. Given a feature space obtained through self-supervised training (see Fig. 2) and a few labeled seed pixels, the open-set prediction module, which can represent several possible algorithms, classifies known pixels and identifies unknown classes, generating the final output.

on remote sensing data, and even fewer methods are suitable for HSI [8], [9]. Due to the highly different characteristics of hyperspectral imagery and the common problems that follow this type of data (small amount of available data, fewer classes, and low openness), it is difficult to directly use open-set techniques developed for daily image recognition [6]. Thus, in this work, we propose a novel framework for HSI pixelwise classification considering both the problem of few available data for training (few-shot learning) and the existence of unknown classes during testing (open-set recognition). Our approach, Few-shot Open-set hyperspectral recognition from Self-Supervised Learning training (FOSSeL), is a generalizable framework that allows for multiple solutions to be implemented on top of the base of the pipeline that performs closed-set predictions from a few labeled samples, similarly to a semisupervised method.

We show that FOSSeL is capable of classifying known classes for an input HSI from a few annotated pixels, while also recognizing as unknown the ones that do not belong to the set of classes of the previously labeled samples. This is done in a pipeline that allows for solutions that do not require supervised training, as the feature space is generated from self-supervised [self-supervised laerning (SSL)] training and the prediction of the pixels is done via label propagation in a graph constructed on top of a super-pixel segmentation, which further allows for the exploitation of the spatial relationship between pixels along with the spectral information. Fig. 1 presents an overview of the proposed FOSSeL pipeline.

While recent works have explored unsupervised representations and graph-based propagation independently, a unified approach that explicitly addresses the OSR in highly constrained few-shot settings remains unexplored. A comprehensive review of the relevant literature, specifically covering open-set recognition and segmentation for both RGB and HSI data, and semisupervised learning, is detailed in Section II. In addition, it is important to note that this study focuses on pixelwise classification in open-set scenarios. The resolution of subpixel fractional abundances (i.e., spectral unmixing of mixed pixels) falls outside the scope of this work.

FOSSeL does not include in its framework the use of the widely employed cross-entropy loss. This is relevant because the performance of methods based on cross-entropy is often limited due to the obtaining of features with low representation, resulting in embedding spaces with weak discrimination capabilities between classes [10]. We exploit and adapt graph-based label-propagation algorithms enforcing spatial connectivity as a means of helping the prediction in an open-set scenario. However, the generalizable aspect of the framework also allows for the implementation of solutions that rely on cross-entropy-based strategies if desired.

We evaluated FOSSeL on three hyperspectral datasets with different characteristics, showing the robustness of the method under varying circumstances, and surpassing the results achieved by several state-of-the-art methods from the literature. The code for the proposed framework is publicly available online.[1]

The main contributions and innovations of the paper are the following.

1) *Flexible self-supervised open-set framework:* We propose FOSSeL, a unified and flexible framework for few-shot, open-set hyperspectral pixelwise classification that does not require supervised training, and can be adapted to different feature extractors and label propagation strategies.
2) *Unsupervised feature validation:* We demonstrate that self-supervised feature spaces are highly effective for open-set hyperspectral classification, even with extremely limited labeled samples.
3) *Novel spatially-aware propagation:* We introduce spatially-aware graph-based label propagation strategies that jointly exploit spectral similarity and spatial consistency for robust classification.

[1][Online]. Available: https://github.com/matheusbarrosp/FOSSeL

To validate these contributions, we empirically demonstrate that the proposed framework achieves competitive or superior performance compared to fully supervised state-of-the-art methods across multiple datasets under strict few-shot settings.

Unlike prior approaches that rely on supervised training or fixed architectures, the proposed framework decouples feature learning from pixel classification, enabling the use of self-supervised representations combined with flexible graph-based inference. This design makes FOSSeL particularly suitable for real-world remote sensing scenarios where labeled data are scarce and unknown classes are expected during deployment.

The rest of this article is organized as follows. In Section II, we discuss related works in the literature that also tackle the OSR in HSI recognition. Section III presents the FOSSeL pipeline and describes how it functions. In Section IV, we detail the experimental setup, presenting the dataset and the baselines used for comparison. Section V presents the results obtained by FOSSeL. Finally, Section VI, concludes this article.

II. Related Works

HSI pixelwise classification, also known as HSI recognition, is a well-studied, although still hot, topic in the remote sensing area. Mapping land cover classes in satellite imagery is a problem that can be modeled as the supervised classification of each pixel, with the closed-set scenario being the standard assumption. A wide range of solutions has been proposed for this problem in the literature. This includes solutions based on convolutional neural networks [11], [12], graph convolutional networks [13], [14], recurrent neural networks [15], [16], transformers [17], [18], generative adversarial networks (GANs) [19], [20], etc. Despite that, none of these methods are originally suited for open-set scenarios, which is a common and important constraint when considering applying the proposed solutions to real-world problems.

In addition, other problems associated with HSI pixelwise classification exist and have been previously explored in the literature, such as end-member extraction or spectral unmixing. Mixed pixels (pixels that cover multiple ground materials, for example) bring more challenges to the problem, since they may account for more than one of the classes of the dataset. End-member extraction is the task that aims to select a collection of "pure" signature spectra of the materials present in the scene [21]. Although we acknowledge that taking into consideration the pureness of the pixel when trying to solve the open-set HSI pixelwise classification problem could be an important step for more robust results, in this work, we do not consider this aspect of the data. Instead, we will focus on the methodology and evaluation based only on the annotation directly available for each dataset.

A. Open-Set Recognition and Segmentation

The OSR has been explored more extensively for natural RGB image classification. Starting with the formalization in the work of Scheirer et al. (2012) [22], several methods have been proposed since then. Geng et al. (2021) [23] separated the work in the OSR literature into discriminative models and generative models. For discriminative ones, OpenMax [24] was one of the seminal deep-learning-based works, a method that replaces the traditional softmax layer with a new layer that adds an unknown output class to the existing K known categories. From there, different approaches were explored in the literature, usually resorting to statistical modeling of data or its features [25] and often estimating an anomaly score and/or employing extreme value theory strategies [24], [26], [27], [28]. Another trend observed in the literature, reconstruction-based methods [26], [29] leverage the reconstruction error of autoencoders to differentiate known samples from unknown ones [25]. In terms of generative models, approaches, such as G-OpenMax [30] and Neal et al. (2018) [31], aimed to generate open-set samples with the use of GANs as a means of helping the model learn what is different from the known training samples.

Open-set semantic segmentation is a more difficult problem due to its pixelwise approach. Some papers propose solutions for nonhyperspectral remote sensing data [7], [8], [10], [32], [33], which share characteristics with HSI, such as high background complexity, class imbalance, and tiny foreground objects [34]. However, unlike HSI, rich hyperspectral information is unavailable for this type of data, and the imagery usually presents much higher spatial resolutions. Therefore, despite it being possible to adapt the architecture of most of these methods to work with hyperspectral inputs, the fact that the specificity of this type of data is not being exploited may lead to suboptimal results, which is a motivation to seek solutions for HSI.

B. Open-Set HSI Pixelwise Classification

Recently, the OSR for HSI has started to catch the attention of the community. MDL4OW [6] was one of the first methods proposed in the context of open-set HSI pixelwise classification. Following a similar trend to previous reconstruction-based methods [26], [32], MDL4OW simultaneously conducts classification and reconstruction of the input data, where the reconstruction output acts as a tool to help separate unknown classes. Another method following a similar strategy, SSLR [5] employs spatial and spectral latent reconstruction to find unknown classes, classifying unknown samples with the help of a defined threshold.

OCN [35] avoids the need for manual thresholding by following a meta-learning-based strategy. OCN learns to reject outliers by optimizing a cross-entropy loss in meta-training episodes where a subset of known classes is treated as pseudounknowns. SSMLP-RPL [36] is a method that introduces reciprocal point learning (RPL) in the open-set context. Working on feature space learning, RPL aims to push the known classes to the periphery of the space according to reciprocal points in the training phase, with unknown samples composing the extra-category space.

OSC-SCL [37] is a method that employs contrastive learning to constrain spectral–spatial features, so that features of the same class can be tightly clustered in the feature space. Unknown classes are found with a threshold based on the distance to the class anchors. FCPN [38] defines prototypes from the extracted features, and a cosine similarity is used to classify unknown objects. EVML [39] is a method that tries to predict a manually

annotated "coarse class" for unknown regions. For each known land cover class, EVML fits a Weibull distribution based on the support feature distance from the respective prototypes.

HyperCASR [40] is a reconstruction-based method that does not focus on the reconstruction of the original pixels, but instead works on extracted semantic features with the aim of reducing the impact of noise from the input data. Finally, more recently, CACL [41] employs transformer-based methods to model the spatial position information of input tokens. A contrastive technique is used to bring samples from the same class together, while pushing away those from different classes. Unknown samples in the testing phase are predicted with an adaptive segmentation threshold.

Our approach is different from the aforementioned works due to several characteristics. First, FOSSeL is a highly generalizable framework, allowing for different implementations of solutions on top of self-supervised training, which is an advantage when compared to methods that present a fixed architecture. Our framework allows for solutions that do not require any type of supervised learning, which can be useful in scenarios where the available annotated samples are highly limited. Finally, our approach efficiently exploits the spatial information of the input data due to the use of graph-based label propagation techniques, while previous works generally rely only on deep learning-based strategies for leveraging spatial information.

C. Semisupervised Learning

Semisupervised learning is another paradigm that is closely related to the way we approach the open-set HSI problem. Semisupervised learning aims to improve learning performance with a few labeled data inputs by using additional unlabeled samples, exploring the latent patterns of the unlabeled data [42]. A few papers have tackled the open-set recognition problem from a semisupervised perspective.

In [43], the authors propose INL, a model that leverages unlabeled data in the framework by training a rotation prediction branch, and classifying unknown samples by checking their distances to the learned class prototypes. SSB [44] is a method that incorporates pseudolabels with high inlier classifier confidence into the training. The authors train the model with pseudolabels from the inlier classifier whose confidence scores are above a predefined threshold, while the outlier detector consists of one-versus-all binary classifiers. Se-FOSS [45] is a method that utilizes a self-supervised consistency loss in an open-set scenario, leveraging all types of unlabeled data and employing an energy-based score to recognize data belonging to known classes.

FOSSeL can be considered similar to semisupervised learning approaches due to leveraging both unlabeled data and the annotation information of a few labeled samples. However, FOSSeL allows for the implementation of solutions that do not include any type of training with the labeled information from the few annotated pixels, with the only training occurring without labels with a self-supervised strategy. Another point is that semisupervised methods often employ pseudolabel strategies [42]. Due to its generalizable properties, FOSSeL can be adapted to employ this type of strategy in the framework, but this is not required. In essence, the similarities between FOSSeL and semisupervised learning come from the similar nature of self-supervised and semisupervised learning, with more direct implementations of semisupervised strategies being possible to be implemented due to the open nature of the framework.

III. Methodology

As illustrated in Fig. 1, FOSSeL employs a two-stage pipeline whose final result is the classification of all input pixels from the HSI, including the separation of the ones, that are unknown compared to the few available annotated samples. The first stage is a self-supervised training that aims to generate a distributed feature space across the classes of the dataset. The second stage involves classifying the pixels in the input image using a graph constructed over a superpixel segmentation. This can be done without further training by exploiting the spatial relationships among superpixels to propagate the labels of known samples to the remaining unlabeled ones.

A. Self-Supervised Feature Generation

Deep-based supervised learning often requires a high amount of labeled data for training [46]. Remote sensing tasks are usually scarce in terms of annotation due to the nature of the applications, which may require the work of a specialist to correctly perform the labeling process. Thus, SSL methods emerge as a promising solution, as they are capable of avoiding the need for fully annotated large-scale datasets by adopting self-defined pseudolabels as supervision [46]. In the context of hyperspectral data, the use of SSL has been explored recently [47] to the point of showing the capability to outperform traditional supervised learning [48].

The first step of the proposed FOSSeL framework is to perform SSL training with all the available data of the target dataset without the need for any labels. This process has the objective of generating a well-distributed feature space for all the pixels of the input HSI, where a classification algorithm can later perform the prediction. While the FOSSeL framework is modular and can accommodate various SSL architectures, in this work we experimented with the feature extraction module using VICReg [49]. The theoretical justification for this choice is twofold. First, unlike contrastive SSL methods (such as SimCLR [50] and MoCo [51]) that heavily rely on mining negative pairs on large batch sizes, VICReg prevents representation collapse purely through information maximization by an explicit constraint on the variance of the embeddings, which was shown to play the role of a negative term between the vectors [49]. In the context of HSI, defining true negative pairs is usually risky due to low spatial resolution and high interclass similarity. Furthermore, VICReg explicitly regularizes the variance and covariance of the embeddings. The covariance regularization forces the learned variables to be decorrelated, which is particularly beneficial for hyperspectral data where spectral bands are typically highly correlated. This approach makes the method robust even with low-dimensional projectors, offering a good

balance between discriminative representation for few-shot scenarios and computational efficiency [49], serving as a robust baseline to evaluate our proposed label propagation mechanics. Other works in the literature, such as Braham et al. (2022) [48], with a Barlow–Twins method [52], have also achieved results that outperform supervised learning before.

Following VICReg's methodology, let $X \in \mathbb{R}^{H\times W\times C}$ be a HSI with height H, width W, and C spectral bands. At every SSL training iteration, we sample a patch p of size $s \times s$ from the X, i.e., $p \in \mathbb{R}^{s\times s\times C}$. In traditional training, this patch would be used to generate two different views of the same input, however, in this work, we follow the pair sampling approach proposed by Braham et al. (2022) [48] in which another random patch with at least 50% overlap to the first one, $p' \in \mathbb{R}^{s\times s\times C}$, is also selected. The idea is that neighbor-overlapping patches should have similar representations, thus serving as a type of data augmentation that exploits the spatial locality of the patches, introducing a spatial regularity prior to the SSL training [48]. Then, two transformations, t and t', are sampled from a distribution T in order to generate different views of p and p': $x = t(p)$ and $x' = t'(p')$. These views are fed to a shared backbone encoder f that will generate their latent representations $r = f(x)$ and $r' = f(x')$, which will later be used as the final feature for each pixel of the HSI for classification. During the SSL training, these representations are mapped to a projection head h that expands the representations to embeddings, $z = h(r)$ and $z' = h(r')$, where the loss is computed. VICReg's loss function is a weighted average of the invariance, variance, and covariance terms (more details can be found in VICReg's original paper [49]), defined as

$$L(Z, Z') = \lambda s(Z, Z') + \mu[v(Z) + v(Z')] + \nu[c(Z) + c(Z')] \tag{1}$$

where λ, μ, and ν are hyperparameters that define the importance of each term, $s(.)$, $v(.)$, and $c(.)$ represent, respectively, the invariance, variance, and covariance regularization terms, while $Z = [z_1, \ldots, z_n]$ and $Z' = [z'_1, \ldots, z'_n]$ represent batches of embeddings generated by the projection head for n different patches from the HSI.

The transformations applied to the input image are an important step for the training of an SSL model. For FOSSeL, along with the aforementioned pair sampling operation that selects a random neighbor overlapping patch to form a pair with the first randomly chosen one, spatial and spectral transforms are applied to the input images. Considering operations that work in the spatial domain, we employed random horizontal flipping, random vertical flipping, and random rotation by multiples of 90°. As for transformations that work in the spectral domain, we apply Gaussian noise, random band dropping, and random band swapping between adjacent bands, inspired by operations that were previously successfully used in the literature [48].

The backbone encoder f of the SSL model can be designed from scratch or adapted from the many neural network architectures that can be found in the literature. Our implementation of FOSSeL employs as its backbone encoder a simple ResNet18 [53] for its lightweight architecture and reduced number of parameters necessary for training. As for the projection head h, we employ an MLP-like architecture, comprised of a sequence of two fully connected layers followed by a 1-D batch normalization layer and ReLU activation, with a last fully connected layer generating the final embedding. Fig. 2 illustrates the entire SSL training process.

B. Pixel Classification

Once the SSL pretraining is complete, the projection head h is discarded, as the rest of the framework works with the latent representation learned by the backbone encoder f. The idea is that a well-trained encoder will be able to generate a feature space where all the classes are separated and distributed, thus later facilitating the work of the classifier.

For all of the following methods, we consider that a few labeled pixels are provided as seeds to the algorithm. From these seeds, the label will be propagated to the remaining unlabeled pixels according to the chosen algorithm. To prevent ambiguity regarding the role of labeled data in our framework, it is crucial to distinguish between the feature generation and pixel classification phases. The feature generation phase of FOSSeL is strictly unsupervised: the network weights are optimized via SSL entirely without the use of ground-truth labels. Once this phase is complete, the network parameters are permanently frozen. During the subsequent pixel classification phase, supervision information is indeed introduced in the form of a very limited number of labeled seed pixels. However, in this phase, the provided labels act exclusively as starting anchors for the graph propagation algorithms. Thus, the seeds are never used to compute a loss gradient, fine-tune the feature space, or update the model parameters. Therefore, while supervision information is required to define the known classes during inference, the underlying representation learning of the framework remains completely unsupervised.

In the literature, previous works have turned their attention to graph-based frameworks, which allow for addressing the high dimensionality and the scarcity of labeled samples problems in HSI [54], [55]. These approaches have shown promising results by constructing a graph that connects the pixels based on spectral similarity and/or spatial adjacency. For example, the Watershed Classifier (WC) [54] has been shown to work well with HSI [55] by exploiting connectivity patterns and spatial relationships without the need for trainable parameters. Furthermore, a graph-based structure is well-suited for label propagation algorithms, which propagate the label information from a few known seeds (known pixels) to the entire set of pixels to be classified, thus achieving robust classification performance even with minimal training data.

However, working with graphs where each pixel represents a vertex can be computationally demanding, as the number of vertices does not scale well for increasingly bigger inputs. Thus, we employ superpixel segmentation as a preprocessing step for the pixel prediction step (not used during the SSL training), as shown in Fig. 1. This approach reduces the number of vertices in the graph, as each superpixel can be seen as a collection of neighboring pixels that are perceptually and/or spectrally similar.

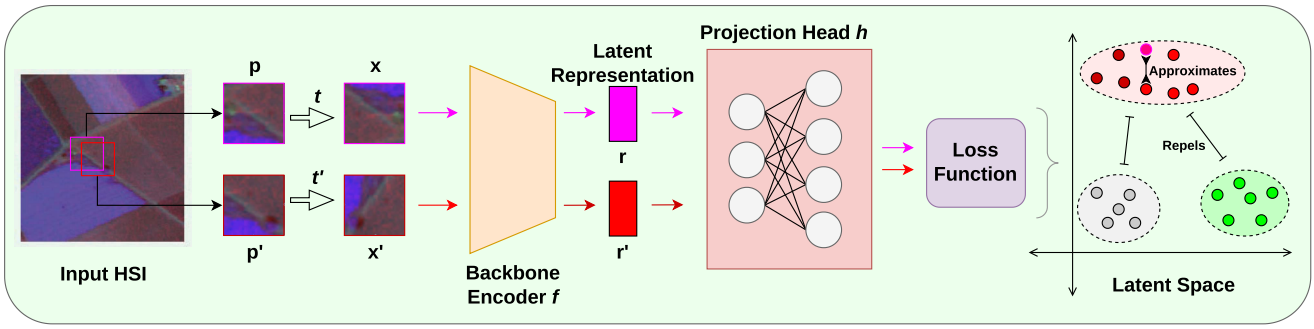


Fig. 2. Overview of the SSL training process. A random patch p and its overlapping neighbor p' are transformed and processed by the backbone encoder to generate the desired representation. During training, the projection head maps the representations to embeddings where the loss function is calculated, which allows for similar samples to be brought together in the feature space, separating them from the dissimilar ones.

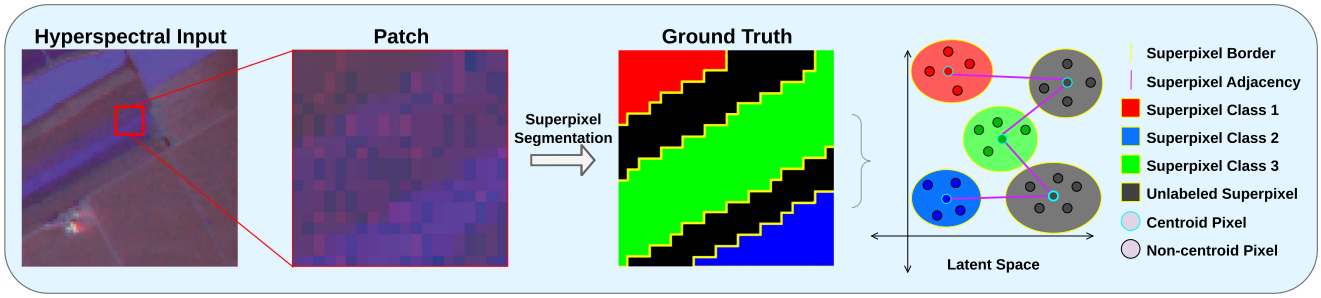


Fig. 3. Illustration of how the graph is constructed from the input HSI and how the pixels are organized in the learned SSL feature space. Each computed superpixel is a vertex and connects to its adjacent neighbors. The centroid pixel of each superpixel is chosen as its representative, being used to compute the edges' weights according to the Euclidean distance in the latent space of centroids to adjacent superpixels.

In the proposed FOSSeL framework, each superpixel represents a vertex in the graph, while the edge set is defined by the superpixels' adjacencies. Thus, each superpixel (vertex) is connected to its adjacent neighbors, with the weight of the edge being defined by the Euclidean distance in the SSL feature space of the centroid pixel (chosen as the "representative" pixel of the superpixel). This way, adjacent vertices, represented by superpixels' centroids, that belong to different classes would ideally be positioned far from each other in the feature space, setting a high value for the edge's weight, while vertices neighboring other superpixels from the same class would be positioned close in the feature space, resulting in a low edge weight. As the SSL training does not use labeled data, these feature distances are learned based on the inherent spectral information of the pixels. Fig. 3 illustrates an example of how this process works. With this approach, the computational cost of the label propagation algorithm is reduced, enabling faster execution times. On top of the computational advantage, the use of superpixels intrinsically incorporates spatial information into the framework. Instead of operating on individual pixels, label propagation occurs between spatially coherent regions, introducing a form of regularization that makes the framework more robust to noise, ensuring that the propagated labels are consistent with object structures and the local spatial context.

As the proposed FOSSeL framework is generalizable and allows for different strategies of pixel (or superpixel) classification to be employed, we selected three different approaches to systematically evaluate the performance of the method with varying graph-based techniques. These methods were selected to represent different paradigms for exploiting the spatio-spectral information of the pixels that is encoded in the graph. The first is a pathfinding algorithm, which classifies a node based on the distance to the closest labeled seed. The second approach is a label propagation model similar to the classic approach proposed by Zhu and Ghahramani (2002) [56], where labels are propagated from a row-stochastic transition matrix constructed on top of the graph's adjacencies. Finally, the last method propagates the labels with a k-Nearest Neighbors (k-NN) approach, considering the representations in the SSL-trained feature space.

One important aspect that is common for all proposed strategies is that for the pixels inside the same superpixel region as a seed, the framework always sets their labels to the same as the respective labeled seed, regardless of the prediction method, as we consider the similarity of these pixels in the superpixel space a stronger evidence of a shared class than any prediction the algorithm may make.

1) Closest Seed Propagation (CSProp) Method: The CSProp method operates on a graph $G = (V, E, W)$ where V represents the set of vertices (superpixels), E represents the set of edges, and $W : E \to \mathbb{R}^+$ represents the weight assigned to each edge (Euclidean distance of centroids' SSL representation).

This algorithm implements a "nearest seed" propagation strategy. First, the set of vertices V is partitioned into two groups: a small subset of seed vertices V_{seed}, for which class labels are known, and the set of unlabeled "test" vertices V_{test}.

The objective of the method is to find the closest seed for every test vertex. To achieve this, a multisource Dijkstra algorithm is executed on the graph, originating from all seed vertices in V_{seed}. This process calculates, for every test vertex $v_j \in V_{\text{test}}$, the shortest path to its nearest seed vertex $v_i \in V_{\text{seed}}$. The label propagation is then performed directly: each unlabeled vertex v_j is assigned the known class label of its corresponding nearest seed v_i. Finally, these newly assigned superpixel labels are mapped back to the original image space, resulting in a thematic map where all pixels within a superpixel region share the same predicted class.

2) Similarity Matrix Propagation (SimProp) Method: This method operates on the principle of information diffusion, aiming to propagate label information from the seed nodes to the remaining ones through the weighted graph, similarly to the method proposed by Zhu and Ghahramani (2002) [56]. First, we convert the weights (previously defined as the Euclidean distance in the feature space between two adjacent nodes) to a similarity weight using the following inverse relationship:

$$w'(e) = \frac{1}{1 + w(e)} \tag{2}$$

where e represents an edge $e \in E$ and $w(e)$ represents its previous weight.

Next, the method builds a similarity graph that is represented as an $N \times N$ weighted adjacency matrix W, where N is the total number of vertices, and the entries W_{ij} correspond to the similarity $w'(e)$ for the edge between nodes i and j. This matrix is normalized to create a row-stochastic transition matrix T, dividing each row of W by its sum

$$T_{ij} = \frac{W_{ij}}{\sum_{k=1}^{N} W_{ik}}. \tag{3}$$

We also construct a label probability matrix P of shape $N \times C$, where C is the number of known classes. This matrix is initialized as follows:

1) *Seed Nodes:* If a node i is a seed of class k, its corresponding row in Y is initialized with the value 1.0 at the kth position and zero in all other positions (one-hot vector); and
2) *Unlabeled Nodes:* All other nodes are initialized with a vector composed of only zeros.

From the transition matrix T and the probability matrix P, the algorithm computes the label propagation. For this, every node gathers label information from its neighbors as the weighted average of its neighbors' current label distributions. This operation is done via a single matrix multiplication

$$P^{(t+1)} = T \times P^{(t)}. \tag{4}$$

After the propagation step, to ensure the seed nodes do not lose their ground-truth information, their label distributions are reset to their original one-hot vectors. This process continues iteratively until the difference between the label distributions from two subsequent iterations falls below a tolerance threshold, or until a maximum number of iterations is reached. The final prediction is set as

$$\text{pred}(v) = \underset{k \in C}{\text{argmax}}(P_{v,k}). \tag{5}$$

3) k-NN Propagation (KNNProp) Method: Unlike the previous methods, the k-NN approach aims to construct a graph where the connections are based on the neighborhood of a node in the feature space, rather than taking into account the original superpixel adjacencies from the input image. For this, an edge is created between a node i and a node j if and only if node j is one of the k-NN of node i in the feature space.

Once the k-NN graph is built, the algorithm iteratively propagates label information similarly to the previous SimProp method via transition matrices. Thus, the main difference between the two approaches lies in the graph itself, more specifically, in the adjacencies that are considered.

C. Opening the Closed-Set Predictions

After executing any of the previously presented label propagation methods, a closed-set classification of the dataset is generated as a result. This can be enough for some applications that are only concerned about recognizing known classes. For the final part of the framework, we explore how to open this closed-set prediction to separate the unknown classes from the rest.

We are interested in performing an open-set classification of the HSI, which means that there are pixels from unknown objects or classes that should not be mislabeled as one of the known annotated classes. Thus, the last question to be answered is: how do we separate the unknown pixels from the ones that belong to the same class as the available annotated seeds? For each one of the three presented methods to propagate the seed labels, we propose an alternative or modification of the algorithm to determine which pixels belong to unknown classes. It is important to note, however, that the previously explained process, from the SSL pretraining to the closed-set prediction, allows for several possibilities to identify unknown classes to be experimented with as well.

One important remark about FOSSeL is that for the proposed approaches, no label is ever provided for unknown pixels, i.e., the user only needs to provide a few seeds for the known classes. It is possible to adapt the method to also take into consideration that the user may label/provide seeds for unknown classes, however, in this work, we only take into consideration the scenario in which the users are interested only in defining what they want to be recognized in the image, rather than having to point toward what they do not want to be recognized as well.

1) Opening CSProp Method: To detect unknown nodes in CSProp, we employ a thresholding technique according to the shortest distances to the seeds. From the closed-set prediction of the previous step, we have access to the value of the distance to the closest seed of node v

$$d_{\text{seed}}(v) = \min_{s \in S} d(v, s). \tag{6}$$

With the shortest distance calculated for each vertex, we have access to a global distribution of seed distances for all nodes

in the graph, $D_{\text{seed}} = \{d_{\text{seed}}(v) \mid \forall v \in G\}$. We then define the OOD threshold, d_{mean}, as the mean of this entire distribution

$$d_{\text{mean}} = \text{mean}(D_{\text{seed}}). \tag{7}$$

Finally, any node v is classified as unknown if its seed distance exceeds this mean threshold.

2) Opening SimProp and KNNProp Methods: Both SimProp and KNNProp employ a similar label propagation algorithm. Their main distinction lies in the graph construction: SimProp connects nodes based on spatial superpixel adjacencies, while KNNProp connects them based on k-NN in the feature space. Thus, the same strategy can be employed to open these two methods.

For these methods, we leverage the closed-set label probability distribution as a measure of classification confidence. Let Y be the final $N \times C$ label probability matrix, where N is the number of nodes and C is the number of classes. For any node v, its probability distribution is the vector $Y_v = [p_1, p_2, \ldots, p_C]$. We define the confidence score for this node, $\text{conf}(v)$, as the maximum probability in its distribution

$$\text{conf}(v) = \max(Y_v). \tag{8}$$

We then establish a minimum confidence threshold, τ. Any node v whose final confidence score is lower than this threshold is classified as unknown

$$\text{pred_open}(v) = \begin{cases} \text{argmax}(Y_v) & \text{if conf}(v) \geq \tau \\ \text{unknown} & \text{if conf}(v) < \tau. \end{cases} \tag{9}$$

The proposed thresholding strategies are intentionally simple and do not require additional training or calibration. Empirically, we observed that these criteria remain stable across datasets and few-shot settings, providing a good balance between correctly classifying known samples and identifying unknown ones. While more sophisticated thresholding strategies could be employed, the adopted approach prioritizes robustness and ease of deployment in practical scenarios.

D. Methodological Novelty

While the FOSSeL framework builds upon foundational concepts in SSL [49] and graph-based semisupervised label propagation [56], it is important to emphasize that our approach is not a trivial concatenation of these methods. Standard SSL methods are not inherently designed for open-set anomaly isolation, and traditional label propagation algorithms frequently fail when applied to the noisy, high-dimensional space of hyperspectral imagery without spatial regularization. The main innovation of FOSSeL lies in formulating a unified framework designed for the few-shot, open-set hyperspectral domain.

IV. Experimental Setup

A. Datasets

We evaluated the FOSSeL framework with three hyperspectral datasets from the literature with different characteristics: Indian Pines (IP), Kennedy Space Center (KSC), and Salinas (SA). For the evaluation of our methods in an open-set scenario, we follow a similar protocol to other works in the literature [5], [6], [7], [10] to adapt datasets that were originally made for closed-set cases. Thus, for each dataset, a subset of classes is selected to be hidden and treated as unknown. This simulates the situation where the remaining classes of the dataset present at least one labeled pixel, therefore being known, while the unknown ones have no label available for training. During the evaluation, we ignore, for all datasets, the pixels that have no label in the ground truth, as we can not determine if they are unknown or not due to the possibility that they contain samples from otherwise known classes that were simply not annotated.

1) Indian Pines (IP): This dataset was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS [57]) sensor over Northwestern Indiana. The scene presents 200 spectral reflectance bands over 145 × 145 pixels, with a wavelength range of $0.4 - 2.5 \times 10^{-6}$ m.

The dataset contains 16 labeled classes: alfalfa, corn no-till, corn min till, corn, trees, pasture, pasture mowed, hay-windrowed, oats, soybeans no-till, soybeans min till, soybeans clean till, wheat, woods, buildings grass trees, and stone-steel towers. From these, three were selected to be hidden and treated as unknown: alfalfa, pasture mowed, and oats. For the choice of the hidden classes, we followed the same set selected in Pal et al. (2022) [35].

2) Kennedy Space Center (KSC): This dataset was also gathered by the AVIRIS [57] over the KSC, Florida, with the same wavelength as IP and from an altitude of approximately 20 km. The scene contains 176 reflectance bands over 512 × 614 pixels.

KSC contains 13 labeled classes: scrub, willow swamp, CP hammock, slash pine, oak/broadleaf, hardwood, swamp, graminoid marsh, spartina marsh, cattail marsh, salt marsh, mud flats, and water. For this dataset, the classes selected to be hidden were scrub, willow swamp, and swamp. The choice for these classes was made by separating the swamp-related classes from the rest, while scrub was additionally selected by randomly picking from the set of classes without a close semantic relation with the others (unlike, for example, the different types of marsh classes).

3) Salinas (SA): The SA dataset is another one that was gathered by the AVIRIS [57] over SA Valley, California. With 204 hyperspectral bands, its spatial resolution is 3.7-m pixels in a scene of size 512 × 217.

This dataset contains 16 labeled classes: broccoli green weeds 1, broccoli green weeds 2, fallow, fallow rough plow, fallow smooth, stubble, celery, grapes untrained, soil vineyard develop, corn senesced green weeds, lettuce romaine 4wk, lettuce romaine 5wk, lettuce romaine 6wk, lettuce romaine 7wk, vineyard untrained, and vineyard vertical trellis. The hidden ones for this scene were the lettuce-related classes, which are lettuce romaine 4wk, lettuce romaine 5wk, lettuce romaine 6wk, and lettuce romaine 7wk. The choice was also based on the set selected by Pal et al. (2022) [35], with the inclusion of the lettuce romaine 5wk, which is set for training in their original GitHub code.

Fig. 4 shows the thematic map of each dataset, illustrating the entire set of labeled pixels for the three scenes and coloring in black the pixels that were not labeled (thus not participating

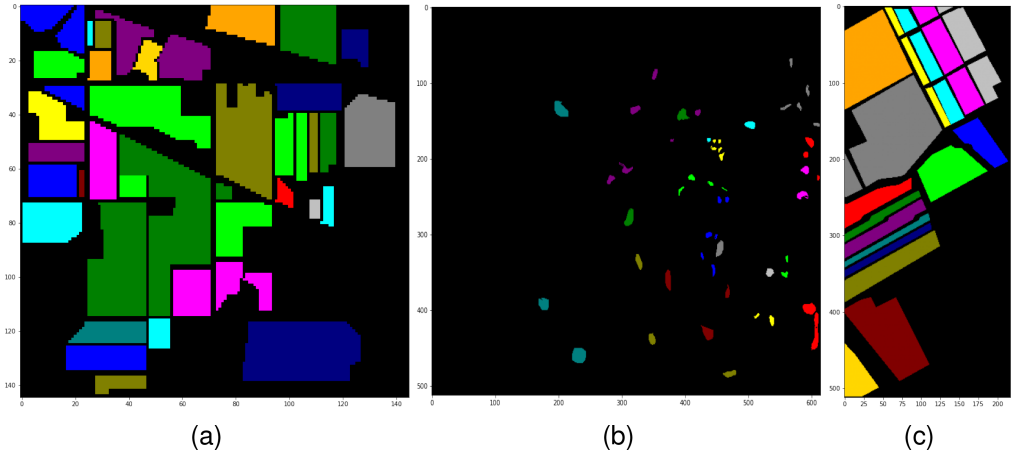


Fig. 4. Visualization of the thematic map of the three datasets. (a) IP. (b) KSC. (c) SA.

in the metrics' calculation). As the figure shows, the datasets present different spatial characteristics beyond the different sets of labeled classes previously detailed. These varying characteristics help to show how well FOSSeL can operate in different scenarios.

B. Implementation Details

FOSSeL allows for different models to be employed in its framework, including different architectures for the encoder backbone of the SSL training and different methods of open-set classification at the final stage. For our experiments, we used the PyTorch framework [58], and the specification of our implementation of the methods is as follows.

We employed VICReg [49] as our core model for the self-supervised training of the framework with the Lightly API[2]. For the SSL loss presented in (1) we set the hyperparameters λ, μ, and ν as 25.0, 25.0, and 1.0, respectively. Regarding the backbone encoder architecture, ResNet18 [53], we adapted PyTorch's torchvision implementation by changing the number of input channels of the first layer to the number of hyperspectral bands of the respective dataset. Thus, the encoder starts by performing a dimensionality reduction on the data by reducing the number of channels to 64 in the first layer. From there, each block in the ResNet18 architecture doubles the number of filters being learned up to 512, which is the output of the encoder, both for the projection head and for the pixel classification in the second stage of the framework. Finally, the projection head, which follows the MLP-like architecture described in Section III-A, increases the number of features to 2048, which is the final size of the embeddings where the loss is computed.

All three datasets were trained for 20 000 epochs with a batch size of 4096 on one GeForce GTX Titan Xp GPU with 12 GB of memory. We experimented with different patch sizes and learning rate values for each one of the datasets, and the best results were achieved with the following configurations: for IP, patch size 11 and learning rate 0.04; for KSC, patch size 9 and learning rate 0.01; and lastly, for SA, patch size 11 and learning rate 0.1. The optimizer used in the training was the Stochastic Gradient Descent.

The rationality of the selected learning rates and patch sizes was based on the stability of the loss convergence along with the standard deviation of the ℓ_2-normalized output ($\ell_2\text{std} = z/||z||_2$, where z is the latent embedding vector), as defined by Chen and He (2021) [59] for siamese-based neural network architectures, as tracking this standard deviation provides a reliable metric to prevent representation collapse. We also visually evaluated the discriminative quality of the resulting feature space by inspecting the distribution of the features according to a t-Distributed Stochastic Neighbor Embedding (t-SNE) [60] plot. We focused our experiments on the IP dataset initially, as it is the smallest among the three. Thus, for this dataset, we experimented with learning rate values varying between 0.01 and 0.1 and with patch sizes varying between 3, 5, 7, 9, and 11. For the smallest patch size (3) with 0.1 learning rate, the training ended with a loss value oscillating around 12.86 and $\ell_2$std of 0.037 (approaching the theoretical optimum of $1/\sqrt{d} \approx 0.044$, where d represents the dimensionality of z). Other learning rate configurations for this patch size either failed to converge (such as with a 0.5 learning rate, which caused *nan* values) or exhibited slow convergence (such as 0.01 and 0.05 learning rates, which stopped with an $\ell_2$std value of 0.034). In contrast, utilizing the largest patch size (11) with a 0.04 learning rate yielded a loss of approximately 16.59, while maintaining an equivalent $\ell_2$std of 0.037. Although the total loss was higher for the largest patch, this is an expected outcome, as the expanded spatial context makes it harder for VICReg's loss invariance term to force augmented views of the same patch into identical representations. Because the $\ell_2$std indicated comparable representation quality across the best cases for each patch size, the final selection of size 11 was driven by the t-SNE feature visualizations [as seen in Fig. 5(a)], since the increased spatial context provided by the larger patch size generated more compact and well-separated feature clusters.

Building upon the baseline established by IP, we evaluated the SA dataset by testing variations of the patch size of 9, 11, and 13,

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

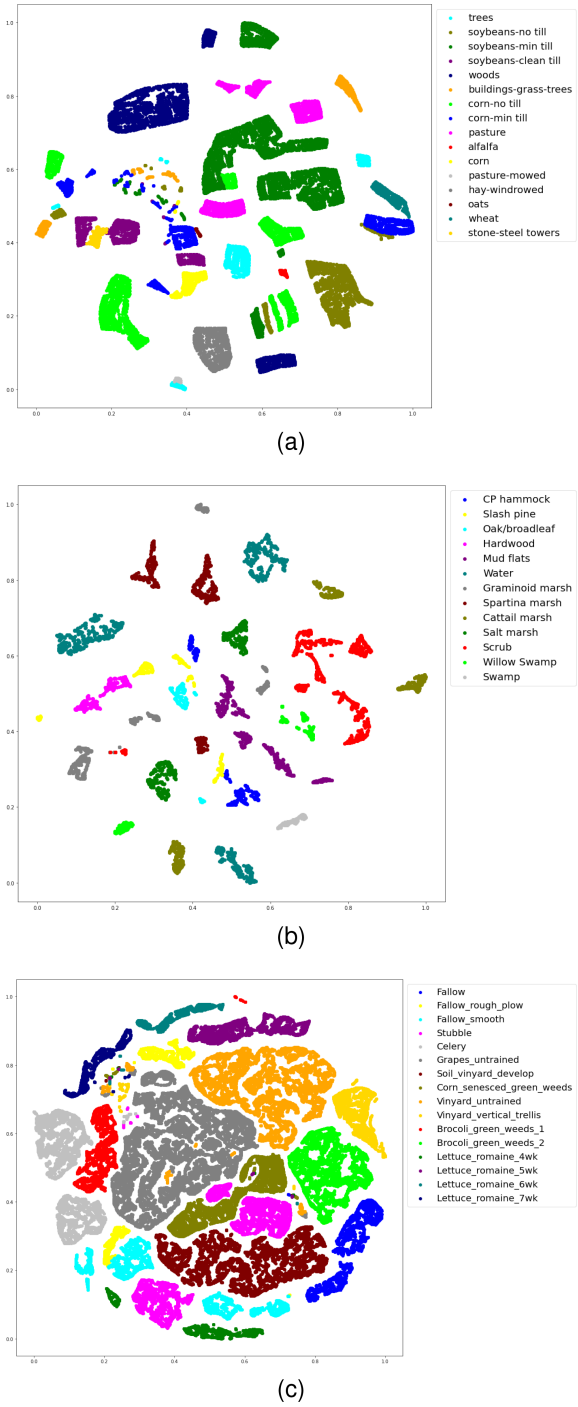


Fig. 5. Feature spaces learned for the known classes via self-supervised training (with VICReg [49]) for all three datasets. Although the classes are identified in the plot for contextualization, no annotation is used in the SSL training. (a) IP. (b) KSC. (c) SA.

alongside learning rates of 0.01, 0.05, and 0.1. For this dataset, a 0.1 learning rate achieved the best results, yielding a mean $\ell_2$std of 0.037 (against the 0.044 theoretical optimum) across the varying patch sizes, whereas the configurations with the lowest learning rate (0.01) achieved a mean $\ell_2$std of 0.035. The best result was ultimately achieved using a patch size of 11, resulting in an $\ell_2$std of 0.0375 and a loss value of 14.11. As for the KSC dataset, learning rates exceeding 0.05 caused training instability and poor convergence. With the lowest experimented learning rate of 0.01, similar results were observed for patch sizes 9, 11, and 13, in which all of the three achieved $\ell_2$std of 0.035. Reducing the patch size to 7 showed a decrease in performance, with $\ell_2$std of 0.033. Given that KSC possesses the highest spatial resolution among the datasets, the final choice between patches 9, 11, and 13 prioritized the option with the more efficient configuration in terms of memory consumption, i.e., patch size 9. Finally, we also conducted additional ablation experiments aimed at increasing the network's focus on the central pixel of the patches. This included omitting the overlapping patch augmentation strategy and artificially assigning a higher weight to the central pixel prior to the forward pass. However, both of these strategies proved detrimental, as they impaired model convergence and resulted in cluttered feature spaces. Table I shows the results of these experiments, with the variations being applied to the baseline configuration of the IP dataset (patch size 11 and 0.04 learning rate).

TABLE I
RESULTS OF THE ABLATION STUDY OF SSL TRAINING STRATEGIES ON THE IP DATASET

Learning Rate	Patch Size	Overlap	Central Pixel Weight	Loss	ℓ_2 std
0.04	11	yes	none	16.56	0.037
0.04	11	no	none	4.86	0.036
0.04	11	no	2.0	4.95	0.036
0.04	11	no	3.0	3.15	0.033
0.04	11	no	4.0	3.19	0.032

"Overlap" denotes the application of the overlapping neighbor augmentation, while "central pixel weight" indicates the scaling factor applied to the central pixel of the patch to evaluate the impact of spatial biasing.

As the results in Table I show, when the overlapping neighbor augmentation strategy is removed ("Overlap: no"), the training loss drops significantly from 16.56 to 4.86. While a lower loss traditionally indicates better convergence, in this context, by applying transformations to the exact same patch rather than overlapping neighbors, the network can easily minimize the loss without learning robust spatial–spectral representations. Conversely, the overlapping strategy introduces an important challenge to the training that forces the network to reconcile spatial shifts, which ultimately yielded a more organized and separated feature space, as seen in Fig. 5(a) along with the higher $\ell_2$std value from the table. Furthermore, the results show that artificially increasing the weight of the central pixel did not enhance the overall feature distribution. Between not using an additional weight to the central pixel ("Central Pixel Weight: none") and using a factor of 2.0, the $\ell_2$std was not considerably affected, while further increases in the factor made the representation deteriorate. By heavily biasing the central pixel, the network effectively ignores the surrounding spatial context and reduces the objective to a simple pixel-to-pixel mapping, which prevents the encoder from capturing important spatial–spectral topologies required by the framework for the following label propagation. Therefore, the results confirm that treating all pixels within the patch equally, in conjunction with the overlapping augmentation strategy, allows for the learning

of the more organized representations needed for an accurate open-set pixelwise classification.

For the superpixel segmentation, we used the simple linear iterative clustering (SLIC) method [61]. As SLIC is designed for 3-channel color spaces, we first applied principal component analysis to the HSI input, reducing it to its three most significant components. This 3-component image was then used as the input for the SLIC segmentation. For all experiments, we configured the SLIC algorithm to generate approximately 1000 superpixels, except for the experiments on the KSC dataset, where we generated approximately 5000 superpixels due to the higher spatial resolution of the input image. Section V-D presents an analysis regarding the impact of the number of segments on FOSSeL's results. A compactness parameter of 15 was used in all experiments.

Finally, regarding the confidence score threshold τ utilized by *SimProp* and *KNNProp*, its optimal strictness is related to the availability of labeled seeds. To empirically validate this thresholding strategy and demonstrate the adaptable tradeoff between known class accuracy and unknown class rejection, we present threshold tradeoff curves in Section V-D. As detailed in this subsequent analysis, while extreme few-shot scenarios operate optimally near a permissive $\tau \in [0.5, 0.6]$, scenarios with more available data work better with stricter thresholds to preserve open-set integrity. Thus, we set the threshold as 0.6 for the scenarios with five and ten seeds, while for the scenarios with 15 seeds we set the threshold as 0.7.

C. Baselines

To evaluate the performance of our proposed FOSSeL framework, we establish comparisons against two different categories of baseline methods. The first category consists of a graph label propagation algorithm from unsupervised training based on the WC [54] and the superpixel-guided label propagation (SGLP) method from DSCA-Net [62]. The WC shares our core methodological paradigm, relying on unsupervised training prior to graph-based label propagation, while the SGLP was designed to propagate known labels via a superpixelwise similarity measure. Thus, both methods serve as direct comparative baselines to validate the efficacy of our propagation strategies. The second category consists of state-of-the-art HSI pixelwise classification algorithms that operate in a supervised manner. Including these supervised models, allows us to evaluate how closely our unsupervised feature space can compete with methods that have the distinct advantage of actively adapting their representations using the labeled data.

1) Graph-Based Unsupervised Propagation Method: For the first category, we compare our results to the WC [54] and the SGLP [62] method. Following the definition of Challa et al. (2022) [54], the WC works on an edge-weighted graph defined in the same way as in the CSProp method. The algorithm assumes a dissimilarity measure $\rho(x, y)$ between two vertices $x, y \in V$ called *pass value*, which is defined as

$$\rho(x, y) = \min_{\pi \in \Pi(x,y)} \max_{e \in \pi} W(e) \tag{10}$$

where π represents a path between the vertices x and y, and Π represents the set of all possible paths between them. Thus, the pass value represents the minimum of the highest weights across the possible paths from x to y.

Once the pass value is defined, by taking the information of a few labeled pixels (seeds) for each known class of the dataset, one can perform a closed-set classification by propagating the label of the seeds to the remaining unlabeled pixels. To do this, for each unlabeled pixel in the set $V_u \subset V$, we calculate the pass value to all the labeled pixels $V_l \subset V$ (where $V_l \cap V_u = \emptyset$) and set the target unlabeled pixel $v \in V_u$ class to the same class as the seed with the minimum pass value

$$\text{pred}(v) = \text{class}(\operatorname{argmin}_{l \in V_l} \rho(v, l)) \tag{11}$$

where class(.) represents a function that returns the label of the respective vertex.

In order to open the WC results, we can treat each seed point as unknown if its class does not exist. Hence, given a set of k known classes $C = \{1, 2, \ldots, k\}$, for each class $i \in C$ we calculate the minimum pass value from the seeds of i, V_l^i, to the seeds of all the other classes $j \in C | j \neq i$, V_l^j

$$p_i = \min_{x \in V_l^i, y \in V_l^j} \rho(x, y) \tag{12}$$

where p_i is the minimum pass value for the class i to all other known classes j.

From the p_i calculated for each class, for a test pixel $v \in V_u$ to be classified as known for class i, its pass value that defines its closed prediction to be class i is expected to be smaller than p_i. Thus, if it is higher, we classify this pixel as unknown

$$\text{pred_open}(v) = \begin{cases} \text{pred}(v) & \text{if } \min_{l \in V_l} \rho(v, l) < p_i \\ \text{unknown} & \text{otherwise} \end{cases} \tag{13}$$

where pred(v) is the closed-set prediction defined (11).

As for SGLP [62], this method was designed to mitigate the negative impact of imbalanced data distribution by leveraging spatial–spectral homogeneity. The process begins by segmenting the input HSI into a set of nonoverlapping superpixels. From a few labeled pixels, the unlabeled superpixels are classified according to a superpixel-level nearest neighbor algorithm. For this, the method calculates the cosine similarity between the average spectral signature of the unlabeled superpixel and the mean spectrum of each class of labeled samples, subsequently assigning the label of the most similar class to the entire superpixel. In the original paper [62], SGLP is employed to enable the equal selection of unlabeled samples from each category, aiming to generate a balanced training set for subsequent closed-set training. In this work, however, we focus on using this method as an open-set label propagation strategy to compare the results with FOSSeL's variants. Thus, to adapt SGLP's strategy into the open-set scenario, a confidence-based decision boundary is introduced on top of the cosine similarity propagation: the maximum similarity score is extracted for each target superpixel, which is then evaluated against a dynamically calculated threshold defined as the overall mean of the maximum similarity scores across all superpixels. If a specific superpixel's highest

similarity fails to exceed this threshold, the closed-set prediction is rejected, being classified as an unknown region.

2) HSI Pixelwise Classification Supervised Methods: To evaluate the proposed FOSSeL framework, we select four recent, state-of-the-art baseline methods specifically designed for open-set HSI pixelwise classification. These baselines comprise a variety of recent architectures and learning paradigms.

1) *OSC-SCL [37]:* A method utilizing supervised contrastive learning designed for HSI open-set scenarios.
2) *SSMLP-RPL [36]:* A spectral–spatial MLP-like network that employs reciprocal points learning to bound known classes for HSI data.
3) *HyperCASR [40]:* A recent spectral–spatial HSI open-set recognition model based on category-aware semantic reconstruction.
4) *CACL [41]:* A state-of-the-art class-aware consistency learning framework specifically designed for open-set semisupervised HSI pixelwise classification.

All the tests were conducted using the authors' original GitHub code for each method, with the only modifications being related to specific characteristics of the experiment, such as defining a new set of hidden unknown classes and the number of available training samples, along with adding support for new datasets that were not originally included.

For OSC-SCL [37], we set the η hyperparameter to 3.5 for the IP and KSC datasets, while for SA, we set it to 2.0. The remaining hyperparameters were set to the same default values defined by the authors. However, it is important to note that, for the KSC dataset, the method was not capable of learning valid anchors for some of the known classes, as it did not correctly predict any sample. Thus, in order to avoid invalid anchors, we used the logits of all available samples for the classes that fell into this situation, instead of just the ones that were classified correctly.

For HyperCARS [40], we set the same hyperparameter values as suggested by the authors on the dataset that was also evaluated in their work, IP. For KSC and SA, the patch size was set to 19, and the number of Groups in the GPE Module to 4.

Finally, regarding SSMLP-RPL [36] and CACL [41], we used the default hyperparameter values proposed by the respective authors, with the only differences being the aforementioned set of unknown classes and the number of training samples.

V. Results and Discussion

This section details the empirical results of our experiments. We analyze the results in four separate parts. First, we present the performance of the SSL stage, which serves as the feature extraction backbone. Second, we compare the open-set HSI pixelwise classification results of FOSSeL with the WC [54], which is another method operating on the same methodological base. Third, we evaluate and discuss the results compared to state-of-the-art supervised methods for HSI open-set pixelwise classification. Lastly, we evaluate the sensitivity of our framework regarding the main hyperparameters previously introduced, such as the number of superpixel segments and the confidence threshold τ.

Overall, the results demonstrate that FOSSeL achieves consistent performance across datasets with different characteristics, maintaining strong accuracy even in extremely few-shot scenarios. Notably, the framework remains competitive with supervised approaches despite not relying on labeled data for training, highlighting the effectiveness of the learned feature space and the proposed propagation strategies.

A. Feature Space Learning

Generating an adequate feature space can be the key to good performance in the prediction phase. This is especially true if the pixel prediction algorithm that follows the SSL training does not require further training, such as the proposed approaches, because the learned feature space is the foundation of the method in those cases. Fig. 5 shows the t-SNE [60] plot of the labeled data for the three evaluated datasets after the SSL training with VICReg [49], where each point represents a labeled pixel of the input scene. The t-SNE representation was computed based on the output of the encoder backbone of the SSL method, whose dimension is of size 512, as detailed in Section IV, instead of the output of the projection head that is used only for loss calculation during the SSL training.

Fig. 5 shows an important outcome of the self-supervised training: the labeled classes tend to form separated clusters, a desirable characteristic to minimize misclassifications later in the pipeline. It is important to note that this class separation was achieved without any annotation. Therefore, the model organized this feature space using only the intrinsic spectral information of each pixel, combined with the local spatial context from its patch neighbors. The plots in Fig. 5 focus on showing the overall organization of only the labeled pixels from the input HSI. However, in practice, the unlabeled pixels usually distribute themselves around the clusters of known classes, either for being other samples from known classes that were not labeled, or for being a sample of an unknown class that could not be grouped with other clusters. This property can be seen in Fig. 6, where the unlabeled pixels are represented by the light-gray points. This aligns with the physical reality of remote sensing, as these intermediate points likely represent boundary pixels or mixed pixels created by the sensor's limited spatial resolution. Consequently, while the labeled class clusters are usually distinct, these surrounding unlabeled points create a smooth transition between them. This illustrates the main challenge of the task: defining a robust decision boundary that can navigate this dense and continuous transition region.

As the three datasets present classes whose pixels are usually contained in continually annotated clusters, this results in clusters of classes better separated in the feature space. It is important to note that datasets, such as IP and SA present classes, with close spatial proximity and distant semantic proximity that were well separated in the feature space nonetheless. An example of this in the IP dataset can be seen with the woods (dark blue) being separated in the feature space [see Fig. 5(a)] from the classes corn-no till (light green), soybeans-no till (olive), and soybeans-min till (dark green), which are spatially close [as viewed on the top right quadrant of Fig. 4(a)]. For SA, this can

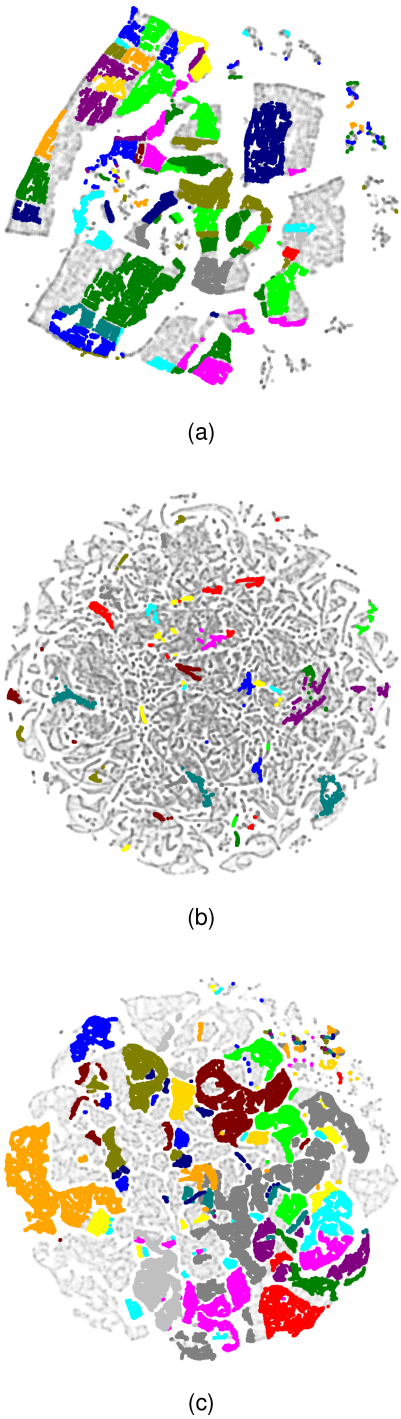


Fig. 6. Feature spaces learned using all pixels, labeled or not, via self-supervised training (with VICReg [49]) for all three datasets. Each light-gray point represents an unlabeled pixel. (a) IP. (b) KSC. (c) SA.

be seen with the class corn senesced green weeds (olive) being separated in the feature space of Fig. 5(c) from the classes lettuce romaine 7wk (navy/darker blue) and soil vineyard develop (dark red), although being spatially close, as seen in Fig. 4(c) (below the middle of the image). This indicates that FOSSeL can work well even in scenarios where one needs to classify all pixels of the input image.

B. Comparison to Unsupervised Graph-Propagation-Based Baselines

From the results obtained by the SSL training for the generation of the feature space, FOSSeL performs pixel prediction from a few labeled pixels for each known class. In Tables II, III, and IV, we show the comparison between the results obtained by FOSSeL and the baseline WC [54], with and without employing the proposed superpixel strategy along with the label propagation algorithm. *OA Open* represents the overall accuracy of the model considering the existence of unknown classes, *OA Closed* represents the accuracy of the model evaluated only on the known classes, *BA Open* is the balanced accuracy considering the presence of the unknown class, *BA Closed* is the balanced accuracy over the known classes only, and *Kappa* is the kappa score, including the unknown class. In addition, in open-set scenarios, these traditional metrics can be limited in the amount of information they bring about the results, particularly when there is a significant imbalance between the number of known and unknown pixels. A desirable approach for an open-set method is to maintain a good balance between correctly classifying the known pixels and detecting unknown samples. Thus, we also adopt the H-score [63] as a main performance indicator. The H-score calculates the harmonic mean between the classification accuracy of known classes and the accuracy of unknown classes, similar to the classic F1-Score, but adapted for the context of OSR. The H-score is defined as follows [63]:

$$\text{H-score} = 2 \cdot \frac{ACC_{\text{known}} \cdot ACC_{\text{unknown}}}{ACC_{\text{known}} + ACC_{\text{unknown}}}. \tag{14}$$

The harmonic mean heavily penalizes large disparities between the two values: known accuracy and unknown accuracy. Consequently, a high H-score can only be achieved if the model successfully maintains high discriminative power for known classes without sacrificing its ability to identify unknown pixels. This prevents scenarios where a model artificially inflates its accuracy by overclassifying pixels into known categories. Thus, we focus our analysis on the H-score, as it provides a balanced metric to evaluate the alterations in the results according to hyperparameter changes.

All the following results were computed, respectively, by using only five, ten, and 15 labeled pixels for label propagation. For each case, we ran the experiments ten times, thus, we also report the metrics in terms of their standard deviation.

As Tables II, III, and IV show, the WC and SGLP consistently achieve worse results than the two best options among the proposed methods, with the WC ranking at best as the second highest option in terms of H-Score in three of the nine experiments. Considering WC, this is directly tied to its criteria when calculating the *pass value*. Because the WC determines the *pass value* based on the minimum of the highest weights between vertices along a path, it frequently encounters the same bottleneck edges for different classes. This limitation results in a high tendency to generate identical pass values for different competing classes, leading to ambiguity and reduced discriminative power during the final classification step.

TABLE II

COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS, THE WC [54], WC WITH SUPERPIXELS (WC W/SP), AND SGLP [62] IN A FIVE-SHOT SCENARIO, WHERE ONLY FIVE LABELED SEEDS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Score
IP	WC	28.21 ± 3.27	27.85 ± 3.34	35.91 ± 3.35	33.48 ± 3.47	23.96 ± 3.17	38.44 ± 4.07
	WC w/sp	44.64 ± 3.96	44.56 ± 4.06	56.60 ± 2.70	56.80 ± 2.34	39.85 ± 3.99	46.52 ± 7.76
	SGLP	51.44 ± 4.99	51.49 ± 4.97	60.02 ± 4.49	61.13 ± 4.10	46.45 ± 5.14	45.07 ± 16.63
	CSProp (ours)	63.63 ± 2.05	63.87 ± 2.00	72.88 ± 1.97	75.57 ± 1.58	60.21 ± 2.10	43.29 ± 22.22
	SimProp (ours)	43.86 ± 2.90	43.36 ± 2.92	62.19 ± 2.13	59.48 ± 2.35	40.99 ± 2.76	59.86 ± 3.38
	KNNProp (ours)	41.31 ± 3.62	40.85 ± 3.65	58.40 ± 2.83	55.92 ± 3.06	38.39 ± 3.48	55.96 ± 4.55
KSC	WC	76.94 ± 2.21	73.14 ± 2.26	67.81 ± 2.62	65.51 ± 3.24	73.27 ± 2.42	80.78 ± 3.53
	WC w/sp	68.34 ± 1.78	86.37 ± 2.49	74.88 ± 1.87	82.14 ± 2.16	65.15 ± 1.97	4.50 ± 2.94
	SGLP	56.92 ± 5.86	72.31 ± 7.45	62.11 ± 4.33	68.32 ± 4.77	52.62 ± 6.51	0.00 ± 0.00
	CSProp (ours)	90.31 ± 1.52	97.43 ± 1.97	93.61 ± 1.38	96.55 ± 1.55	89.10 ± 1.70	77.38 ± 1.45
	SimProp (ours)	88.09 ± 2.77	85.61 ± 3.36	84.06 ± 1.54	82.75 ± 1.65	86.19 ± 3.22	90.99 ± 2.19
	KNNProp (ours)	79.67 ± 1.36	81.35 ± 1.62	74.18 ± 1.06	74.25 ± 1.33	76.52 ± 1.55	77.02 ± 3.22
SA	WC	66.06 ± 2.85	67.62 ± 1.48	62.31 ± 3.58	63.28 ± 2.49	62.18 ± 3.23	54.60 ± 15.82
	WC w/sp	84.01 ± 1.97	84.34 ± 2.19	86.50 ± 1.70	86.98 ± 1.85	82.18 ± 2.20	82.49 ± 1.00
	SGLP	67.64 ± 5.11	70.21 ± 5.53	72.11 ± 2.88	74.59 ± 3.12	63.94 ± 5.58	52.70 ± 3.32
	CSProp (ours)	85.47 ± 2.19	87.13 ± 2.24	87.35 ± 1.45	88.88 ± 1.47	83.84 ± 2.40	76.93 ± 3.81
	SimProp (ours)	70.08 ± 2.38	69.51 ± 2.52	72.60 ± 1.48	72.34 ± 1.46	66.82 ± 2.56	72.20 ± 4.08
	KNNProp (ours)	86.58 ± 4.06	86.53 ± 4.33	88.34 ± 1.59	88.44 ± 1.66	85.03 ± 4.43	86.69 ± 3.72

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

TABLE III

COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS, THE WC [54], WC WITH SUPERPIXELS (WC W/SP), AND SGLP [62] IN A TEN-SHOT SCENARIO, WHERE ONLY TEN LABELED SEEDS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Score
IP	WC	37.00 ± 2.29	36.78 ± 2.42	43.80 ± 1.47	42.46 ± 1.86	32.67 ± 2.06	45.03 ± 2.63
	WC w/sp	56.61 ± 3.01	56.68 ± 3.00	68.26 ± 2.10	69.67 ± 1.93	52.46 ± 3.17	49.97 ± 13.35
	SGLP	61.46 ± 3.42	61.66 ± 3.44	68.46 ± 3.34	70.68 ± 3.07	57.26 ± 3.61	43.27 ± 21.83
	CSProp (ours)	74.19 ± 2.67	74.59 ± 2.69	80.18 ± 1.94	83.94 ± 1.78	71.44 ± 2.81	41.36 ± 17.50
	SimProp (ours)	64.21 ± 3.17	63.94 ± 3.18	76.71 ± 2.17	75.48 ± 2.15	61.33 ± 3.18	75.36 ± 5.06
	KNNProp (ours)	60.96 ± 3.21	60.72 ± 3.22	74.19 ± 2.04	73.27 ± 1.89	57.96 ± 3.19	70.41 ± 7.39
KSC	WC	82.33 ± 1.49	78.49 ± 1.48	74.03 ± 1.52	71.82 ± 1.65	79.43 ± 1.70	86.40 ± 1.81
	WC w/sp	71.21 ± 0.00	90.71 ± 0.00	78.27 ± 0.00	86.10 ± 0.00	68.42 ± 0.00	0.00 ± 0.00
	SGLP	63.73 ± 3.49	80.97 ± 4.43	70.93 ± 2.66	78.02 ± 2.92	60.13 ± 3.87	0.00 ± 0.00
	CSProp (ours)	91.30 ± 0.03	98.51 ± 0.05	94.44 ± 0.04	97.38 ± 0.05	90.21 ± 0.03	78.32 ± 0.02
	SimProp (ours)	90.56 ± 1.27	89.51 ± 1.62	85.95 ± 0.65	85.11 ± 0.71	89.10 ± 1.47	91.89 ± 0.87
	KNNProp (ours)	80.42 ± 0.00	83.21 ± 0.00	75.42 ± 0.00	75.94 ± 0.00	77.45 ± 0.00	76.18 ± 0.00
SA	WC	67.43 ± 1.58	70.25 ± 1.20	64.95 ± 1.86	67.06 ± 1.49	63.77 ± 1.80	49.62 ± 8.59
	WC w/sp	85.12 ± 4.29	89.66 ± 1.44	88.38 ± 3.32	92.37 ± 0.83	83.44 ± 4.76	43.08 ± 41.83
	SGLP	71.45 ± 4.10	74.41 ± 4.52	75.09 ± 2.09	77.83 ± 2.31	68.16 ± 4.45	53.81 ± 2.27
	CSProp (ours)	91.86 ± 1.59	93.58 ± 1.43	93.95 ± 1.09	95.53 ± 1.04	90.93 ± 1.76	83.15 ± 3.81
	SimProp (ours)	83.96 ± 1.59	87.00 ± 1.63	85.73 ± 1.35	88.37 ± 1.38	82.12 ± 1.75	66.62 ± 2.93
	KNNProp (ours)	91.44 ± 1.87	92.67 ± 2.06	93.05 ± 0.88	94.19 ± 0.97	90.44 ± 2.06	85.44 ± 3.06

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

TABLE IV

COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS, THE WC [54], WC WITH SUPERPIXELS (WC W/SP), AND SGLP [62] IN A 15-SHOT SCENARIO, WHERE ONLY 15 LABELED SEEDS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Score
IP	WC	42.65 ± 3.21	42.42 ± 3.33	49.80 ± 1.57	48.45 ± 2.62	38.34 ± 3.20	51.11 ± 3.03
	WC w/sp	64.81 ± 3.38	65.10 ± 3.35	74.46 ± 2.15	77.57 ± 1.59	61.26 ± 3.62	43.43 ± 10.78
	SGLP	67.24 ± 2.38	67.56 ± 2.33	73.92 ± 2.83	77.11 ± 1.98	63.62 ± 2.58	38.81 ± 22.35
	CSProp (ours)	79.32 ± 1.87	79.68 ± 1.76	84.52 ± 1.93	87.87 ± 0.82	77.05 ± 2.01	52.19 ± 15.44
	SimProp (ours)	65.68 ± 2.10	65.41 ± 2.13	78.41 ± 0.80	77.16 ± 0.91	62.90 ± 2.08	77.28 ± 2.16
	KNNProp (ours)	64.45 ± 2.09	64.14 ± 2.11	77.53 ± 1.24	76.04 ± 1.27	61.59 ± 2.11	77.09 ± 2.21
KSC	WC	85.20 ± 1.03	82.04 ± 0.95	77.97 ± 1.22	76.12 ± 1.27	82.79 ± 1.18	88.67 ± 1.25
	WC w/sp	71.18 ± 0.00	90.70 ± 0.00	78.27 ± 0.00	86.10 ± 0.00	68.39 ± 0.00	0.00 ± 0.00
	SGLP	66.80 ± 2.38	84.86 ± 3.02	75.16 ± 2.03	82.68 ± 2.23	63.52 ± 2.65	0.00 ± 0.00
	CSProp (ours)	91.29 ± 0.00	98.49 ± 0.00	94.42 ± 0.00	97.36 ± 0.00	90.19 ± 0.00	78.33 ± 0.00
	SimProp (ours)	88.73 ± 0.00	86.30 ± 0.00	82.39 ± 0.00	80.87 ± 0.00	86.90 ± 0.00	91.59 ± 0.00
	KNNProp (ours)	80.42 ± 0.00	82.76 ± 0.00	74.89 ± 0.00	75.19 ± 0.00	77.38 ± 0.00	76.93 ± 0.00
SA	WC	68.12 ± 1.09	71.93 ± 0.70	66.25 ± 1.13	69.22 ± 0.91	64.55 ± 1.17	41.86 ± 11.65
	WC w/sp	84.70 ± 3.11	91.65 ± 1.15	88.20 ± 2.46	94.18 ± 0.81	82.96 ± 3.46	17.90 ± 33.73
	SGLP	74.26 ± 3.63	77.47 ± 3.99	77.42 ± 2.32	80.32 ± 2.56	71.26 ± 4.00	54.88 ± 2.19
	CSProp (ours)	94.43 ± 0.60	96.39 ± 0.57	96.04 ± 0.52	97.78 ± 0.46	93.78 ± 0.67	84.41 ± 1.74
	SimProp (ours)	86.03 ± 1.36	87.41 ± 1.47	88.73 ± 1.06	90.09 ± 1.17	84.42 ± 1.50	79.16 ± 1.80
	KNNProp (ours)	92.98 ± 1.33	92.70 ± 1.40	94.66 ± 0.77	94.57 ± 0.82	92.16 ± 1.47	94.18 ± 1.15

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

As for SGLP, its comparatively poor overall performance can be primarily attributed to its reliance on the original hyperspectral data space for label propagation. Because SGLP computes similarities directly on the original pixel data rather than on a pretrained feature space, it can hardly deal with complex spectral–spatial relationships. This limitation strongly reinforces the importance of employing FOSSeL's SSL pretraining. Despite lacking this pretraining step, SGLP shows an overall performance similar to the WC: on the IP dataset, its results align with the WC w/sp variant, and it also achieves overall superior results compared to both WC versions on the SA dataset. However, the absence of a refined, self-supervised feature space proves critically detrimental on the KSC dataset, where the algorithm completely fails to accurately identify any unknown pixel, yielding an H-score of 0 across all evaluated cases.

A comparative analysis between the standard WC and its superpixel-enhanced variant (WC w/sp) illustrates the advantage of performing some type of spatial regularization in HSI pixelwise classification. As Tables II, III, and IV show, the standard WC achieves poor results across the majority of evaluated scenarios when considering the BA Open metric as the main indicator, particularly within the IP and SA datasets. However, the addition of the superpixel segmentation (WC w/sp) brings a performance improvement in these cases. For example, in the SA 15-shot scenario, the BA Open increases from 66.25% with the standard WC to 88.20% with WC w/sp. This difference shows that pixelwise graph propagation can be susceptible to local noise and spectral variability inherent to hyperspectral data. Hence, the superpixel segmentation step acts as a regularizer, reducing the effect of local anomalies, and establishing a more balanced (and smaller) graph topology. This observation validates our framework's reliance on superpixel segmentation to stabilize the underlying graph before propagation begins. However, this strategy can also worsen the performance, as seen in WC for the KSC dataset, especially in cases where neighboring classes get combined in a single superpixel, misclassifying part of the pixels from the group depending on the centroid's prediction. On the other hand, by considering the H-Score as the main metric, we can see that the standard pixelwise approach consistently yields more balanced results between known and unknown accuracy for ten- and 15-shot scenarios. This is especially evident in the KSC dataset, where H-Score drops to zero in those cases, indicating that no unknown pixel was correctly isolated, whereas the known classes are being classified correctly, which can be observed with the higher BA Closed metric values. This reinforces the fact that superpixel segmentation can compromise the classification when neighboring pixels from different classes get combined in the same segment.

Unlike the WC, which can have its results heavily compromised in superpixel-based graphs due to pass-value bottlenecking, the proposed methods demonstrated robust performance across all datasets and few-shot configurations. Furthermore, comparing our approach to SGLP highlights the important role of the SSL pretraining for open-set prediction. Because SGLP relies on original raw data for propagation, it often fails to separate complex classes effectively from the unknown regions. In contrast, our method operates on a more refined, self-supervised feature space that improves prediction robustness. These results highlight the importance of combining rich feature representations with spatial regularization in hyperspectral pixelwise classification. By grouping pixels into coherent regions, superpixels reduce the impact of local spectral variability and noise, enabling more stable label propagation. At the same time, this introduces a tradeoff, as incorrect segmentation may propagate errors within a region.

In addition, we can also assess the computational efficiency gained by the superpixel segmentation phase. For this, we compared the inference times of the proposed superpixel-based graph against a standard pixel-level graph. We evaluated the most computationally demanding scenario: the highest resolution dataset (KSC) under the 15-shot configuration. Operating strictly at the pixel level generates a massive graph comprising 314 368 nodes (as it is an image of resolution 512×614) and 627 610 edges, which drastically inflates the parameter quantity during adjacency matrix construction. When instantiated using the NetworkX Python library [64], this pixelwise graph consumes approximately 360 MB of memory. While this memory consumption is well within the limits of modern hardware, the main bottleneck of graph-based label propagation is the time complexity required to traverse all of the interconnected nodes. By contrast, the proposed superpixel-based graph (configured to generate approximately 5000 SLIC [61] segments) compresses the topology to just 4651 nodes and 10 284 edges, which represents a reduction of over 98% in both graph dimensions, also reducing the memory usage to a mere 5.64 MB. This structural compression translates to important reductions in runtime: for the KNNProp method, the pixel-level graph requires approximately 2327 s (almost 40 min) to propagate labels, whereas the superpixel graph executes the same task in just 4 s. Similarly, our proposed CSProp method drops from 13 s on the pixel-level graph to a highly efficient 1.6 s on the superpixel graph. SimProp was the only exception, as it remained computationally bottlenecked at approximately 60 s in both configurations, indicating a heavy algorithmic overhead independent of the graph size. These metrics demonstrate that the superpixel strategy is a necessary methodological mechanism to make graph-based open-set pixelwise classification memory-efficient and computationally viable for high-resolution hyperspectral imagery.

While CSProp establishes itself as the best algorithm across most datasets when focusing on the BA Open metric, by considering the H-Score as the main metric, SimProp and KNNProp consistently rank higher due to better maintaining the known/unknown accuracy balance. This is especially true for the IP dataset, where not only does CSProp rank below the other two FOSSeL variants, but it also shows a high standard deviation for the H-Score. The IP dataset is challenging due to its high concentration of mixed pixels and intraclass variance. In such environments, the explicit spatial-feature exploitation utilized by CSProp may propagate noise. On the other hand, SimProp relies on a smoother propagation, mitigating part of the local ambiguities present in the IP data.

However, the superiority of the CSProp method in terms of BA Open can be explained by the fact that CSProp explicitly

TABLE V
COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS AND THE SUPERVISED BASELINES IN A FIVE-SHOT SCENARIO, WHERE ONLY FIVE LABELED PIXELS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Ccore
IP	OSC-SCL [37]	41.29 ± 2.00	40.96 ± 2.01	55.91 ± 1.95	54.32 ± 1.90	37.70 ± 1.86	53.26 ± 2.72
	SSMLP-RPL [36]	41.49 ± 4.85	41.25 ± 4.85	53.08 ± 4.54	51.57 ± 4.35	38.18 ± 4.53	52.43 ± 5.65
	HyperCASR [40]	56.13 ± 5.20	56.65 ± 5.24	61.60 ± 3.27	66.30 ± 3.49	50.72 ± 5.40	1.01 ± 2.41
	CACL [41]	69.73 ± 3.72	69.79 ± 3.78	75.85 ± 2.49	76.78 ± 2.32	66.71 ± 3.79	65.01 ± 10.61
	CSProp (ours)	63.63 ± 2.05	63.87 ± 2.00	72.88 ± 1.97	75.57 ± 1.58	60.21 ± 2.10	43.29 ± 22.22
	SimProp (ours)	43.86 ± 2.90	43.36 ± 2.92	62.19 ± 2.13	59.48 ± 2.35	40.99 ± 2.76	59.86 ± 3.38
	KNNProp (ours)	41.31 ± 3.62	40.85 ± 3.65	58.40 ± 2.83	55.92 ± 3.06	38.39 ± 3.48	55.96 ± 4.55
KSC	OSC-SCL [37]	35.99 ± 9.85	18.51 ± 12.51	20.52 ± 6.84	12.59 ± 7.50	19.59 ± 12.97	29.38 ± 17.63
	SSMLP-RPL [36]	67.41 ± 5.43	77.31 ± 6.96	68.05 ± 6.40	71.78 ± 7.35	63.21 ± 6.22	40.50 ± 18.07
	HyperCASR [40]	71.04 ± 3.19	88.64 ± 2.49	79.39 ± 2.83	86.66 ± 2.95	68.12 ± 3.40	10.75 ± 15.34
	CACL [41]	81.80 ± 6.57	91.37 ± 2.68	85.84 ± 2.42	89.74 ± 3.16	79.61 ± 7.15	55.34 ± 28.09
	CSProp (ours)	90.31 ± 1.52	97.43 ± 1.97	93.61 ± 1.38	96.55 ± 1.55	89.10 ± 1.70	77.38 ± 1.45
	SimProp (ours)	88.09 ± 2.77	85.61 ± 3.36	84.06 ± 1.54	82.75 ± 1.65	86.19 ± 3.22	90.99 ± 2.19
	KNNProp (ours)	79.67 ± 1.36	81.35 ± 1.62	74.18 ± 1.06	74.25 ± 1.33	76.52 ± 1.55	77.02 ± 3.22
SA	OSC-SCL [37]	60.07 ± 2.65	60.59 ± 3.19	76.38 ± 1.71	78.17 ± 1.84	56.66 ± 2.80	56.49 ± 7.15
	SSMLP-RPL [36]	74.92 ± 5.11	75.68 ± 5.85	81.62 ± 5.25	82.80 ± 6.06	72.16 ± 5.57	69.90 ± 10.39
	HyperCASR [40]	73.58 ± 1.70	80.87 ± 1.90	80.52 ± 2.01	87.08 ± 2.16	70.63 ± 1.84	3.26 ± 2.74
	CACL [41]	80.79 ± 5.06	83.94 ± 4.95	85.33 ± 3.91	88.30 ± 3.89	78.65 ± 5.56	61.05 ± 13.49
	CSProp (ours)	85.47 ± 2.19	87.13 ± 2.24	87.35 ± 1.45	88.88 ± 1.47	83.84 ± 2.40	76.93 ± 3.81
	SimProp (ours)	70.08 ± 2.38	69.51 ± 2.52	72.60 ± 1.48	72.34 ± 1.46	66.82 ± 2.56	72.20 ± 4.08
	KNNProp (ours)	86.58 ± 4.06	86.53 ± 4.33	88.34 ± 1.59	88.44 ± 1.66	85.03 ± 4.43	86.69 ± 3.72

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

TABLE VI
COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS AND THE SUPERVISED BASELINES IN A TEN-SHOT SCENARIO, WHERE ONLY TEN LABELED PIXELS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Score
IP	OSC-SCL [37]	56.05 ± 1.60	55.87 ± 1.60	71.74 ± 1.01	71.50 ± 0.92	53.03 ± 1.55	63.91 ± 2.51
	SSMLP-RPL [36]	55.69 ± 4.06	55.58 ± 4.08	65.53 ± 3.39	64.96 ± 3.72	52.54 ± 3.94	62.77 ± 4.85
	HyperCASR [40]	66.54 ± 3.25	67.14 ± 3.29	71.20 ± 1.73	76.44 ± 1.99	62.37 ± 3.54	5.57 ± 6.16
	CACL [41]	85.59 ± 2.21	85.79 ± 2.18	87.12 ± 2.34	88.93 ± 1.77	83.80 ± 2.43	71.51 ± 13.06
	CSProp (ours)	74.19 ± 2.67	74.59 ± 2.69	80.18 ± 1.94	83.94 ± 1.78	71.44 ± 2.81	41.36 ± 17.50
	SimProp (ours)	64.21 ± 3.17	63.94 ± 3.18	76.71 ± 2.17	75.48 ± 2.15	61.33 ± 3.18	75.36 ± 5.06
	KNNProp (ours)	60.96 ± 3.21	60.72 ± 3.22	74.19 ± 2.04	73.27 ± 1.89	57.96 ± 3.19	70.41 ± 7.39
KSC	OSC-SCL [37]	62.78 ± 5.48	57.14 ± 7.10	48.60 ± 6.75	45.14 ± 7.82	55.29 ± 6.85	66.87 ± 5.77
	SSMLP-RPL [36]	74.85 ± 3.07	90.12 ± 3.70	80.77 ± 3.55	87.03 ± 3.95	72.03 ± 3.45	29.77 ± 8.56
	HyperCASR [40]	74.82 ± 1.23	94.60 ± 1.25	85.61 ± 1.18	93.83 ± 1.26	72.31 ± 1.34	6.42 ± 6.12
	CACL [41]	82.28 ± 4.69	97.49 ± 1.57	90.09 ± 1.97	96.36 ± 2.64	80.35 ± 5.07	37.40 ± 26.16
	CSProp (ours)	91.30 ± 0.03	98.51 ± 0.05	94.44 ± 0.04	97.38 ± 0.05	90.21 ± 0.03	78.32 ± 0.02
	SimProp (ours)	90.56 ± 1.27	89.51 ± 1.62	85.95 ± 0.65	85.11 ± 0.71	89.10 ± 1.47	91.89 ± 0.87
	KNNProp (ours)	80.42 ± 0.00	83.21 ± 0.00	75.42 ± 0.00	75.94 ± 0.00	77.45 ± 0.00	76.18 ± 0.00
SA	OSC-SCL [37]	80.13 ± 2.03	84.73 ± 0.83	88.18 ± 1.51	92.63 ± 0.57	78.03 ± 2.22	45.50 ± 21.61
	SSMLP-RPL [36]	81.50 ± 3.11	82.79 ± 3.84	86.94 ± 1.83	88.46 ± 2.04	79.39 ± 3.39	72.32 ± 17.48
	HyperCASR [40]	76.02 ± 1.73	83.32 ± 1.93	83.67 ± 1.94	90.30 ± 2.09	73.36 ± 1.87	7.66 ± 5.95
	CACL [41]	88.26 ± 2.23	92.78 ± 2.14	91.08 ± 1.45	95.02 ± 1.07	86.91 ± 2.47	57.48 ± 16.13
	CSProp (ours)	91.86 ± 1.59	93.58 ± 1.43	93.95 ± 1.09	95.53 ± 1.04	90.93 ± 1.76	83.15 ± 3.81
	SimProp (ours)	83.96 ± 1.59	87.00 ± 1.63	85.73 ± 1.35	88.37 ± 1.38	82.12 ± 1.75	66.62 ± 2.93
	KNNProp (ours)	91.44 ± 1.87	92.67 ± 2.06	93.05 ± 0.88	94.19 ± 0.97	90.44 ± 2.06	85.44 ± 3.06

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

leverages both the spatial structure of the graph and the feature-space distances to labeled seeds. By relying on shortest paths, the method effectively constrains label propagation to spatially consistent regions while still accounting for spectral similarity, resulting in more robust predictions in few-shot scenarios, especially for the known classes.

KNNProp was also shown to be a strong alternative, particularly on the SA dataset, where it achieves the best results in all of the scenarios. This is the pixel prediction algorithm that relies most heavily on feature proximity within the SSL space rather than explicit spatial information. Thus, its strong performance helps to demonstrate that the unsupervised features can be highly discriminative by themselves, which is particularly valuable for real-world scenarios where no label information may be available. However, the superiority of SimProp and CSProp over KNNProp in other scenarios confirms that working with spatial knowledge along with SSL features tends to yield the most accurate results.

C. Comparison to Supervised Baselines

In Tables V, VI, and VII, we show the comparison between the results obtained by FOSSeL and the supervised baselines OSC-SCL [37], SSMLP-RPL [36], HyperCASR [40], and CACL [41]. The experiments follow a similar pattern to the comparisons with the WC, thus, the results were computed, respectively, by using only five, ten, and 15 labeled pixels for training (in the case of the baselines) or label propagation (in the case of FOSSeL) over ten repetitions.

Regarding the overall performance and dataset-specific behaviors, the quantitative results presented in Tables V, VI, and VII show the robust overall performance of the proposed FOSSeL framework across the varying few-shot scenarios. Our framework, with varying pixel prediction algorithms, consistently ranks among the best-performing methods. Notably, in all scenarios for SA and KSC, two different variants of our framework occupy the top two positions. On the IP dataset, our

TABLE VII
COMPARISON BETWEEN THE RESULTS OF THE PROPOSED FOSSEL METHOD WITH VARYING PIXEL PREDICTION ALGORITHMS AND THE SUPERVISED BASELINES IN A 15-SHOT SCENARIO, WHERE ONLY 15 LABELED PIXELS ARE AVAILABLE FOR EACH KNOWN CLASS

Dataset	Method	OA Open	OA Closed	BA Open	BA Closed	Kappa	H-Score
IP	OSC-SCL [37]	72.42 ± 2.71	72.50 ± 2.72	81.02 ± 1.49	82.31 ± 1.35	69.84 ± 2.81	67.88 ± 5.06
	SSMLP-RPL [36]	63.33 ± 4.56	63.29 ± 4.60	71.64 ± 3.12	71.71 ± 3.60	60.30 ± 4.66	66.42 ± 4.40
	HyperCASR [40]	73.45 ± 2.04	74.07 ± 2.03	76.89 ± 1.50	82.19 ± 1.18	70.14 ± 2.17	13.19 ± 12.09
	CACL [41]	90.81 ± 2.04	91.20 ± 2.10	89.87 ± 1.43	92.94 ± 1.52	89.61 ± 2.27	63.76 ± 8.76
	CSProp (ours)	79.32 ± 1.87	79.68 ± 1.76	84.52 ± 1.93	87.87 ± 0.82	77.05 ± 2.01	52.19 ± 15.44
	SimProp (ours)	65.68 ± 2.10	65.41 ± 2.13	78.41 ± 0.80	77.16 ± 0.91	62.90 ± 2.08	77.28 ± 2.16
	KNNProp (ours)	64.45 ± 2.09	64.14 ± 2.11	77.53 ± 1.24	76.04 ± 1.27	61.59 ± 2.11	77.09 ± 2.21
KSC	OSC-SCL [37]	65.00 ± 4.33	64.18 ± 9.30	53.71 ± 7.34	52.29 ± 9.97	58.54 ± 5.28	60.33 ± 13.05
	SSMLP-RPL [36]	79.52 ± 4.61	92.81 ± 1.43	84.75 ± 2.05	90.21 ± 1.94	77.16 ± 5.00	41.59 ± 22.43
	HyperCASR [40]	77.21 ± 0.72	97.28 ± 1.03	88.34 ± 0.81	96.61 ± 0.99	74.91 ± 0.78	10.63 ± 4.45
	CACL [41]	85.50 ± 5.94	99.12 ± 0.73	93.16 ± 2.85	98.78 ± 0.93	83.90 ± 6.48	48.89 ± 26.67
	CSProp (ours)	91.29 ± 0.00	98.49 ± 0.00	94.42 ± 0.00	97.36 ± 0.00	90.19 ± 0.00	78.33 ± 0.00
	SimProp (ours)	88.73 ± 0.00	86.30 ± 0.00	82.39 ± 0.00	80.87 ± 0.00	86.90 ± 0.00	91.59 ± 0.00
	KNNProp (ours)	80.42 ± 0.00	82.76 ± 0.00	74.89 ± 0.00	75.19 ± 0.00	77.38 ± 0.00	76.93 ± 0.00
SA	OSC-SCL [37]	80.92 ± 1.47	87.16 ± 1.40	87.97 ± 1.09	93.67 ± 0.73	78.85 ± 1.61	30.05 ± 15.78
	SSMLP-RPL [36]	84.14 ± 2.28	85.67 ± 2.67	89.13 ± 1.47	90.82 ± 1.38	82.32 ± 2.52	74.76 ± 11.29
	HyperCASR [40]	79.47 ± 1.49	87.09 ± 1.72	86.70 ± 1.38	93.55 ± 1.43	77.20 ± 1.64	8.35 ± 5.94
	CACL [41]	89.87 ± 1.73	94.74 ± 0.81	92.67 ± 1.36	96.90 ± 0.51	88.69 ± 1.92	56.45 ± 15.09
	CSProp (ours)	94.43 ± 0.60	96.39 ± 0.57	96.04 ± 0.52	97.78 ± 0.46	93.78 ± 0.67	84.41 ± 1.74
	SimProp (ours)	86.03 ± 1.36	87.41 ± 1.47	88.73 ± 1.06	90.09 ± 1.17	84.42 ± 1.50	79.16 ± 1.80
	KNNProp (ours)	92.98 ± 1.33	92.70 ± 1.40	94.66 ± 0.77	94.57 ± 0.82	92.16 ± 1.47	94.18 ± 1.15

The best results in terms of H-Score for each dataset are in bold, and the second-best results are in red.

framework showed itself to be capable of achieving competitive results, securing the second-best performance in terms of H-Score in the five-shot scenario and the best performance on the remaining ones. As previously mentioned, this decline in performance compared to the other two datasets can be inferred to be caused by a high intraclass variance for the dataset, which causes different clusters for the same class to be separated in the feature space, as seen in Fig. 5(a). However, even in the five-shot scenario, our framework achieved considerably better results than the other supervised baselines apart from CACL.

While the CACL [41] baseline achieves better results in specific cases (most notably on the IP dataset), this comparison must be contextualized by the differences in the learning paradigms. It is important to note that our proposed FOSSeL method is trained entirely without supervision, and the final pixel prediction is done without any subsequent fine-tuning or training on the labeled support set. Consequently, our method operates at a natural disadvantage, as the experimental protocol did not allow us to adapt its learned feature space using the explicit knowledge provided by the labeled samples. The fact that FOSSeL achieves highly competitive (and often superior) results under these constraints highlights the representational power of the unsupervised feature space it generates.

Regarding OSC-SCL [37] and SSMLP-RPL [36] methods, the results show representational limits across varying data distributions. These two methods exhibit considerable dataset sensitivity: SSMLP-RPL achieves competitive results for the SA dataset, even surpassing CACL, but is still behind FOSSeL's variants, whereas the results on the KSC dataset demonstrate that the method can not maintain the same level of known/unknown class accuracy balance. OSC-SCL, on the other hand, presents the opposite characteristic by achieving a better balance for the KSC dataset, while not keeping the same performance on SA. HyperCASR consistently achieves the least balanced results, with the overall worst H-Score performance despite competitive results in terms of BA Open, while still not being able to surpass FOSSeL's performance. Without employing spatial-feature clustering techniques as our framework, these methods lack the representational power to fully exploit the limited available set of labeled pixels, reinforcing the necessity of our proposed spatial propagation strategies.

Considering our proposed methods, an interesting behavior is observed within the KSC dataset, where the performance of our variants remains similar between the ten-shot and 15-shot scenarios. This indicates that with just ten labeled samples per class, our method acquires sufficient information to get close to its performance ceiling. This rapid saturation is likely due to the characteristics of the KSC dataset, which naturally facilitate the efficient exploitation of spatial information, allowing our propagation algorithms to converge early without needing additional labeled pixels. Also, an overall trend of diminishing returns is evident across all datasets. For our proposed methods, the improvement in prediction accuracy from the five-shot to the ten-shot scenario is larger than the subsequent improvement from ten to 15 shots. This shows that the FOSSeL framework requires only a few seeds (approximately ten per class) to accurately map the topology of the classes, especially when the SSL training succeeds best.

Regarding the evaluation of the results in an open scenario versus closed scenarios, the results show FOSSeL's capacity to maintain accurate predictions for the known classes even when the prediction is opened to include unknown classes. As expected, closed-set metrics are higher than open-set metrics across most of the methods and datasets, as the open-set paradigm introduces important challenges to the prediction. However, our proposed methods maintain a close balance between the two, which can be observed due to the overall best performance according to the H-Score. That happens because FOSSeL relies on label propagation through an unsupervised feature space, naturally bounding the known classes based on their intrinsic clustering. Consequently, opening the predictions does not substantially degrade the accuracy of the known classes,

TABLE VIII
COMPUTATIONAL TIME (IN S) FOR TRAINING AND INFERENCE STAGES AVERAGED OVER DIFFERENT FEW-SHOT SCENARIOS FOR THE SAME DATASET

Dataset	Method	Training Time (s)	Inference Time (s)
IP	OSC-SCL [37]	259	12
	SSMLP-RPL [36]	483	75
	HyperCASR [40]	141	8
	CACL [41]	4179	4
	CSProp	$\approx$ 120000	0.1
	SimProp	$\approx$ 120000	4
	KNNProp	$\approx$ 120000	0.3
KSC	OSC-SCL [37]	176	6
	SSMLP-RPL [36]	336	36
	HyperCASR [40]	154	6
	CACL [41]	3971	47
	CSProp	$\approx$ 120000	1
	SimProp	$\approx$ 120000	45
	KNNProp	$\approx$ 120000	2
SA	OSC-SCL [37]	243	64
	SSMLP-RPL [36]	830	705
	HyperCASR [40]	178	66
	CACL [41]	4824	17
	CSProp	$\approx$ 120000	0.6
	SimProp	$\approx$ 120000	8
	KNNProp	$\approx$ 120000	0.8

The training times for the proposed methods encompass the full, unlabeled self-supervised phase.

which is extremely important when employing these methods in real-world settings.

Finally, the evaluation across the three datasets shows that no single label propagation variant is universally optimal. Instead, their performance depends on the scenario and the characteristics of the datasets.

1) CSProp enforces a direct, greedy assignment, according to the closest labeled seed. This characteristic is better for the correct classification of known categories, which can be seen by how it achieves the highest open balanced accuracy (BA Open) overall. This comes from a fundamental characteristic of CSProp, as it is a deterministic, parameter-free approach to open-set recognition. Rather than requiring an empirically tuned confidence threshold to separate known from unknown pixels, CSProp establishes a dynamic decision boundary based on the mean distance to the labeled seeds. While this mechanism simplifies the operational pipeline by removing a hyperparameter, it introduces a strict spatial assumption that can become a limitation depending on the underlying feature distribution. In contrast, the SimProp and KNNProp variants utilize explicitly defined confidence thresholds. This parameterized approach provides flexibility, allowing for granular control to balance the tradeoff between known-class prioritization and unknown-class isolation.
2) SimProp's row-stochastic transition matrix approach prevents overconfidence by ensuring that the propagation flow is probabilistically distributed. This creates the best balance between classifying known targets and rejecting unknown ones, which is empirically validated by SimProp achieving the highest H-Scores on the IP and KSC datasets. However, this global diffusion inherently dilutes the label probability at semantic boundaries and regions combining multiple classes. For a pixel to be classified as a known class in SimProp, its confidence must surpass an explicitly defined threshold. Thus, ambiguous pixels frequently fail to meet this threshold and are classified as unknown. This explains why SimProp usually can not classify known pixels as well as CSProp (which can be observed by comparing their closed metrics), but ultimately achieves the most balanced open-set detection.
Conversely, KNNProp distinctively exploits the distribution of the self-supervised feature space by connecting vertices based on their proximity in the feature space, rather than relying exclusively on spatial pixel connections. The advantage of this localized propagation is that it directly exploits the intrinsic clustering quality of the SSL encoder. Consequently, KNNProp can be the best choice in scenarios where the dataset produces a highly separable and well-distributed feature space, such as the SA dataset [as seen in Fig. 5(c)]. In these scenarios, the nearest neighbor constraint allows for the propagation to accurately occur inside the clusters without diluting label confidence across boundaries. However, this dependency also means that KNNProp's efficacy is strictly bound to the robustness of the SSL representation, as poorly separated feature spaces will cause misclassifications. This can be observed for the KSC dataset, where KNNProp achieves the overall worst results across the three proposed variants.

1) Time Cost Analysis: To provide a comprehensive assessment of the framework's operational practicality, Table VIII presents a quantitative comparison of the computational time costs for both the training and inference stages for the baselines and FOSSeL's variations. As observed in the table, the proposed framework requires a significantly higher training time (approximately 120 000 s) compared to baseline methods like HyperCASR or OSC-SCL. This disparity is a direct consequence of the architectural design. Our 120 000-s training duration represents an exhaustive SSL phase, completely independent of label annotation. While computationally intensive, this offline pretraining is a one-time cost that eliminates the need for dense human annotations to construct a robust feature space.

On the other hand, FOSSeL's inference time, which is the most critical metric for real-world operational deployment, is the lowest in all cases, except for SimProp in the KSC dataset. Once the feature space is established, the proposed CSProp method operates with exceptional speed. For instance, on the SA dataset, CSProp requires only 0.6 s for complete label propagation, vastly outperforming baselines, such as SSMLP-RPL (705 s) and OSC-SCL (64 s). A similar efficiency is observed on the KSC dataset, where CSProp executes in just 1 s, compared to 47 s for CACL.

These results show that the proposed framework heavily frontloads its computational burden into the offline, self-supervised training phase, but it yields fast inference, which is highly desirable for rapid open-set pixelwise classification upon deployment.

Despite its strong performance, the proposed framework presents some limitations. In particular, the reliance on superpixel segmentation introduces sensitivity to segmentation

TABLE IX
NUMBER OF GENERATED SUPERPIXELS (SEGMENTS) AND THE PERCENTAGE OF PURE SUPERPIXELS, DEFINED AS THOSE WHERE ALL LABELED PIXELS BELONG TO A SINGLE KNOWN CLASS

Dataset	SLIC Segments	Generated Segments	Pure Segments %
IP	500	536	87
	1000	793	90
	2000	2247	97
KSC	2500	2463	98
	5000	4651	98
	10000	8207	100
SA	500	436	90
	1000	869	95
	2000	2190	98

Superpixels consisting entirely of unlabeled background pixels are excluded from this calculation.

quality, which can affect performance when neighboring classes are grouped together, as explored in the following Section V-D. In addition, the open-set prediction relies on simple thresholding strategies, which, although effective in practice, could be further improved with adaptive or learned criteria. These aspects represent promising directions for future work.

D. Analysis of Sensitivity to Hyperparameters

To evaluate the robustness and stability of the proposed framework, this section presents a sensitivity analysis of its two main hyperparameters previously introduced: the number of generated superpixels (number of SLIC segments, in this case) and the confidence score threshold τ. To assess how these hyperparameter variations impact the tradeoff between closed-set and unknown pixel accuracy, we utilize the H-score as our primary performance indicator.

1) Number of Segments: To evaluate the robustness of the FOSSeL framework regarding its spatial contextualization via superpixels, we conducted the sensitivity analysis with varying granularities: 500, 1000, and 2000 segments for the IP and SA datasets, and 2500, 5000, and 10 000 segments for the KSC dataset.

To isolate the impact of spatial granularity, we conducted the evaluation of the number of segments using a fixed, baseline confidence threshold of 0.5 across all seed configurations for SimProp and KNNProp. While a 0.5 threshold is well adapted for a highly constrained five-shot scenario, it is less ideal when more labels are available (in this case, with ten or 15 seeds). However, applying an optimized, highly restrictive threshold at this stage would act as an aggressive outlier filter, masking the underlying topological vulnerabilities caused by a suboptimal segmentation. By maintaining a permissive baseline threshold that accepts the closed-set classification of low-certainty samples, we can better observe the impact that the number of segments imposes on the label propagation process.

The results of the sensitivity analysis regarding the number of segments can be seen in Fig. 7. Table IX complements these results by showing how many superpixels were created according to each experiment. The results demonstrate that increasing the number of superpixels does not universally improve performance. Rather, oversegmentation can lead to degradation of the H-score. In datasets characterized by lower spatial resolution, such as IP, using too few superpixels can limit the framework's discriminative capacity. For example, at 500 segments in IP, the boundaries between classes become too coarse, frequently resulting in undersegmentations where distinct classes, including the unknown one, get combined within a single superpixel, negatively impacting the performance. Consequently, an intermediate granularity of 1000 segments emerges as the most stable configuration for IP across the different few-shot scenarios.

For the SA dataset, performance peaks at 1000 segments before degrading at higher granularities. This degradation exposes a fundamental vulnerability in graph-based label propagation related to oversegmentation: as the number of superpixels increases, the distinctions between neighboring regions become less apparent, effectively blurring the natural boundaries that separate different classes. During graph propagation, this boundary degradation causes labels to erroneously "leak" from known seeds into unwanted unknown regions with a confidence higher than the baseline threshold. This vulnerability can also be observed in the KSC dataset: while intermediate segmentations provide balanced results, an oversegmentation (such as with 10 000 segments) severely impacts algorithms like KNNProp, since almost all unknown pixels are being misclassified as known classes due to the label leakage, dropping the H-score to near zero. With a higher confidence score of 0.7, KNNProp's H-score results on KSC would increase to 55% with ten seeds, for example, which is considerably more balanced in terms of known and unknown classes' accuracy. These observations highlight one of the advantages of the proposed CSProp method, as it does not require additional hyperparameter tuning, remaining stable across varying scenarios.

Table IX details the relationship between the spatial granularity and the resulting node purity. As expected, increasing the number of generated segments boosts the percentage of pure superpixels, culminating in values close to or equal to 100% purity in all datasets. However, correlating to the previous discussion, maximizing individual node purity through heavy oversegmentation can be detrimental. This happens because although these highly fragmented nodes are internally homogeneous, their reduced footprint blurs the transitions between distinct regions, allowing for labels to erroneously leak across the graph. Therefore, the empirical data validate that intermediate segmentations that secure sufficiently high purity (e.g., 90%–95%) provide the optimal balance.

2) Confidence Threshold: Having established the spatial granularity for each dataset (1000 segments for IP and SA, and 5000 segments for KSC), the next phase of our evaluation investigates the impact of the confidence score threshold τ employed by the SimProp and KNNProp methods. This threshold dictates how strict the open-set rejection mechanism is. To assess how this parameter regulates the tradeoff between classifying known classes and identifying unknown pixels, we varied the threshold from 0.4 to 0.8 across all three few-shot scenarios. The resulting H-scores are presented in Fig. 8.

As shown in the figure, overly permissive thresholds (e.g., 0.4) consistently lead to a severe performance degradation in open-set recognition. At these lower confidence levels, the model classifies almost all unknown pixels as known targets due to

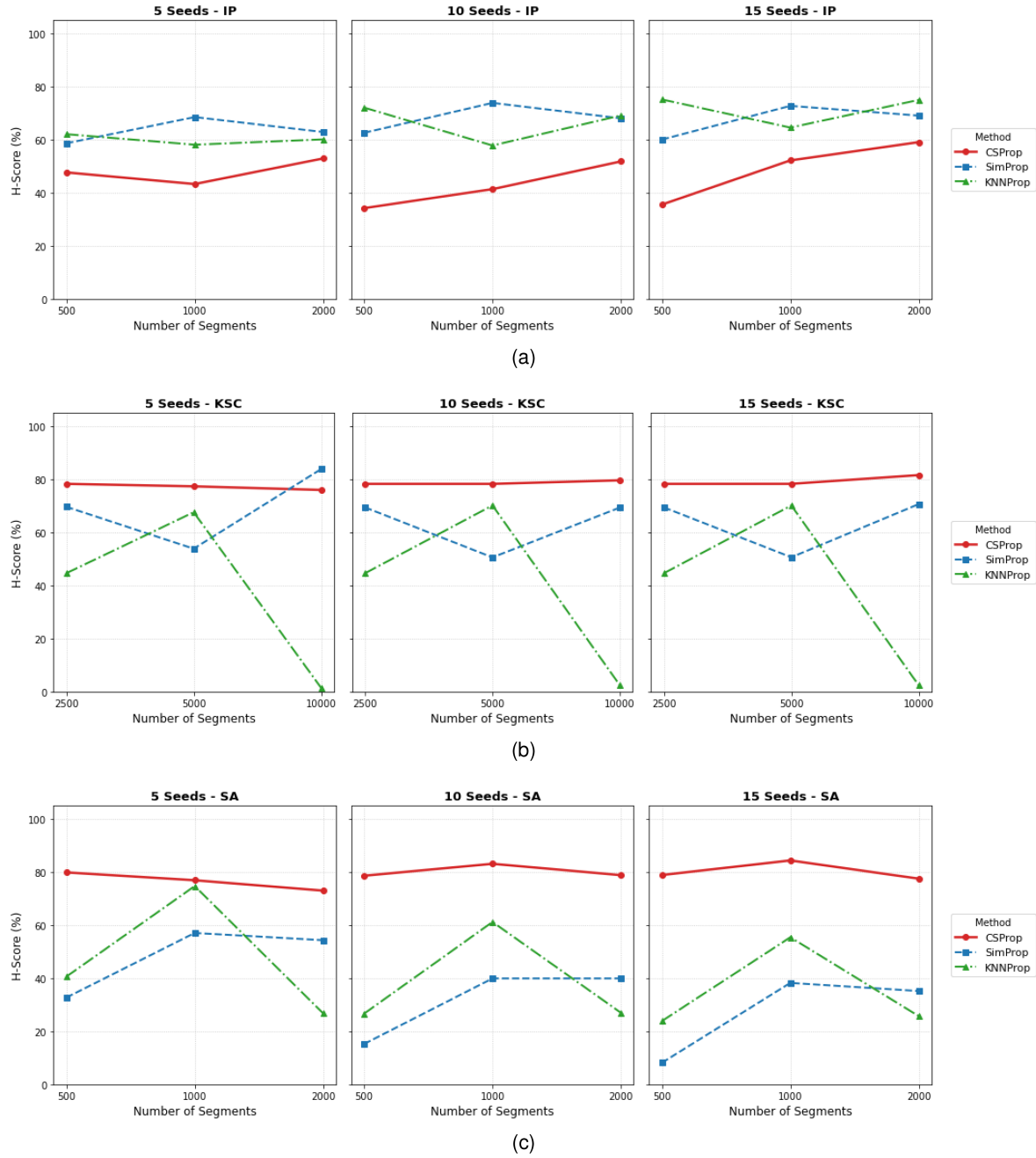


Fig. 7. FOSSeL's performance in terms of H-Score according to variations on the number of generated superpixel segments and the number of available seeds. (a) IP. (b) KSC. (c) SA.

the low confidence needed to assign a label. This is particularly problematic for KNNProp, as it makes its H-score drop toward zero in multiple scenarios.

The results reveal that the optimal confidence threshold is dependent on the abundance of labeled data. In extreme few-shot scenarios (such as the five-seed configuration), the overall confidence propagated through the graph is inherently lower. Thus, employing a threshold between 0.5 and 0.6 provides the optimal balance. Conversely, as the availability of labeled seeds increases in the ten- and 15-shot scenarios, the framework requires stricter thresholds, with the best performances consistently shifting toward the τ values of 0.6 and 0.7.

The data also demonstrates that pushing the threshold further to higher values (e.g., 0.8) makes it so that the known classes' predictions become too difficult, causing pixels from known classes to be erroneously classified as unknown. This explains the start of the decline in H-scores observed at the far right of the plots across nearly all configurations.

E. Fairness of Baseline Comparison and Practical Implications

An important aspect of our evaluation is the asymmetric nature of comparing FOSSeL, which relies entirely on an

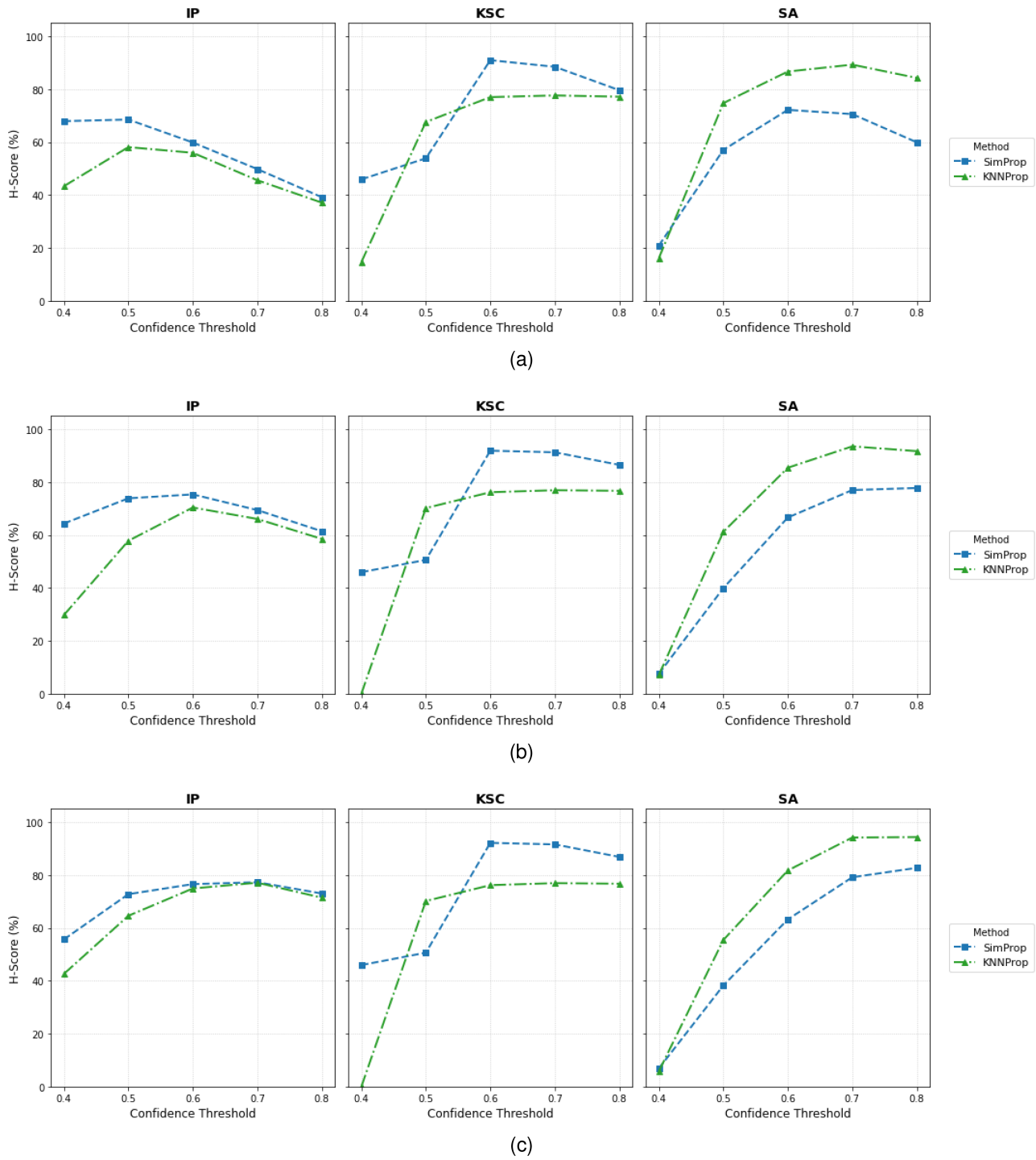


Fig. 8. Performance of SimProp and KNNProp, in terms of H-Score, as the confidence threshold τ varies from 0.4 to 0.8 for each few-shot scenario. (a) Five seeds. (b) Ten seeds. (c) 15 seeds.

unsupervised feature space, against supervised baselines, such as CACL [41], HyperCASR [40], SSMLP-RPL [36], and OSC-SCL [37]. We acknowledge that this comparison presents inherent fairness issues, as supervised methods possess the advantage of actively adapting their feature representations using the labeled support set. Conversely, FOSSeL operates at a natural disadvantage, functioning as a "supervision-free" classification framework, utilizing the few labeled seeds strictly for graph-based label propagation without any backpropagation or fine-tuning. However, we argue that this comparison serves to rigorously benchmark our framework. The empirical results demonstrate that FOSSeL frequently surpasses these supervised baselines in highly constrained few-shot scenarios (e.g., five-shot). Furthermore, while the deployment of our method still requires a minimal set of annotated seeds for the final prediction phase, evaluating the capabilities of unsupervised methodologies holds significant scientific value. It proves that unannotated hyperspectral data can be effectively

structured and clustered prior to any human annotation, serving as a robust foundation that may inspire future advancements in semisupervised and zero-shot remote sensing research for OSR.

VI. Conclusion

In this article, we propose FOSSeL, a framework that performs open-set HSI pixelwise classification based on the feature space generated by a SSL method. We showed how SSL, in our case represented by the VICReg [49] method, can learn how to generate an adequate feature space for HSI without the need for labeled samples. We proposed three different pixel prediction algorithms that can be employed at the end of the FOSSeL framework, which shows how the framework is generalizable and allows for different approaches according to the characteristics of the problem, even without the need for further training or fine-tuning. Through the evaluation across three distinct hyperspectral datasets, we demonstrated that unsupervised feature learning, coupled with superpixel segmentation and label propagation from a few seeds, can effectively compete with, and frequently surpass, supervised baselines.

The pixel prediction methods explored in this work do not require many labeled samples, which is an important characteristic for dealing with HSI problems, known for being data-scarce. The framework exhibited a high efficiency in few-shot scenarios evaluated under the open-set paradigm, achieving a competitive performance with as few as five or ten labeled samples per known class. FOSSeL also demonstrated statistical stability and robustness across diverse datasets without negatively impacting the accuracy of the known classes. However, we note that as the availability of labeled data increases, supervised baselines can eventually surpass our framework, especially on specific datasets where the SSL generates a less organized feature space.

The results indicate that combining self-supervised feature learning with spatially aware graph-based inference is a promising direction for addressing real-world hyperspectral classification challenges, particularly in scenarios with limited annotations and unknown classes. Because FOSSeL is designed as a highly modular framework, a promising avenue for future research involves a rigorous empirical benchmarking of various SSL architectures. Exploring how different self-supervised paradigms and their respective hyperparameter configurations influence the structure of the feature space will provide deeper insights into optimizing the subsequent open-set label propagation phase.

Finally, future work will explore how FOSSeL can be expanded to deal with domain adaptation problems, with the objective of using information from different data sources compared to the target dataset, such as from a different geographical location. We also plan to expand the use of FOSSeL to non-hyperspectral remote sensing data, which is a type of data with different characteristics that would require different techniques for the SSL training and pixel prediction methods. Allowing FOSSeL to work under these circumstances would amplify its use in multiple scenarios, such as cases in which the dataset presents a higher spatial resolution, and even datasets with more fine-grained details for the classes.

References

[1] B. Hosseiny, M. Mahdianpari, M. Hemati, A. Radman, F. Mohammadimanesh, and J. Chanussot, "Beyond supervised learning in remote sensing: A systematic review of deep learning approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 1035–1052, 2024.

[2] M. K. Khlifi, W. Boulila, and I. R. Farah, "Graph-based deep learning techniques for remote sensing applications: Techniques, taxonomy, and applications—A comprehensive review," *Comput. Sci. Rev.*, vol. 50, 2023, Art. no. 100596.

[3] L. Fang et al., "Open-world recognition in remote sensing: Concepts, challenges, and opportunities," *IEEE Geosci. Remote Sens. Mag.*, vol. 12, no. 2, pp. 8–31, Jun. 2024.

[4] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[5] J. Yue, L. Fang, and M. He, "Spectral-spatial latent reconstruction for open-set hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 31, pp. 5227–5241, 2022.

[6] S. Liu, Q. Shi, and L. Zhang, "Few-shot hyperspectral image classification with unknown classes using multitask deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5085–5102, Jun. 2021.

[7] H. Oliveira, C. Silva, G. L. Machado, K. Nogueira, and J. A. Dos Santos, "Fully convolutional open set segmentation," *Mach. Learn.*, vol. 122, pp. 1733–1784, 2023.

[8] C. C. V. da Silva, K. Nogueira, H. N. Oliveira, and J. A. d. Santos, "Towards open-set semantic segmentation of aerial images," in *Proc. Latin Amer. GRSS/ISPRS Remote Sens. Conf. (LAGIRS)*, 2020, pp. 16–21.

[9] F. Ji et al., "Spectral-spatial evidential learning network for open-set hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5503617.

[10] A. Brilhador, A. E. Lazzaretti, and H. S. Lopes, "A prototypical metric learning approach for open-set semantic segmentation on remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5640114.

[11] D. Kumar and D. Kumar, "A binary grey wolf optimization based hybrid convolutional neural network (BGWOHCNN) framework for hyperspectral image classification," *Multimedia Tools Appl.*, vol. 83, no. 4, pp. 10091–10114, 2024.

[12] D. Wang, B. Du, and L. Zhang, "Fully contextual network for hyperspectral scene parsing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5501316.

[13] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Attention graph convolutional network for disjoint hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, 2024, Art. no. 5503005.

[14] J. Xu et al., "Fuzzy graph convolutional network for hyperspectral image classification," *Eng. Appl. Artif. Intell.*, vol. 127, 2024, Art. no. 107280.

[15] W. Zhou, S.-i. Kamata, Z. Luo, and H. Wang, "Multiscanning strategy-based recurrent neural network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5521018.

[16] W. Zhou, S.-I. Kamata, H. Wang, and X. Xue, "Multiscanning-based RNN–transformer for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5512319.

[17] L. Scheibenreif, M. Mommert, and D. Borth, "Masked vision transformers for hyperspectral image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2166–2176.

[18] X. Zhang, Y. Su, L. Gao, L. Bruzzone, X. Gu, and Q. Tian, "A lightweight transformer network for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5517617.

[19] Z. Yu and W. Cui, "Robust hyperspectral image classification using generative adversarial networks," *Inf. Sci.*, vol. 666, 2024, Art. no. 120452.

[20] A. Qin et al., "Distance constraint-based generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5511416.

[21] J. Plaza, E. M. Hendrix, I. Garcıá, G. Martıń, and A. Plaza, "On endmember identification in hyperspectral images without pure pixels: A comparison of algorithms," *J. Math. Imag. Vis.*, vol. 42, pp. 163–175, 2012.

[22] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

[23] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3614–3631, Oct. 2021.

[24] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1563–1572.

[25] I. Nunes, C. Laranjeira, H. Oliveira, and J. A. dos Santos, "A systematic review on open-set segmentation," *Comput. Graph.*, vol. 115, pp. 296–308, 2023.
[26] P. Oza and V. M. Patel, "C2AE: Class conditioned auto-encoder for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2302–2311.
[27] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4011–4020.
[28] G. Yang, S. Zhou, and M. Wan, "Open-set recognition model based on negative-class sample feature enhancement learning algorithm," *Mathematics*, vol. 10, no. 24, 2022, Art. no. 4725.
[29] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional Gaussian distribution learning for open set recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13477–13486.
[30] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 42.1–42.12. [Online]. Available: https://bmvc2017.london/, https://dblp.org/db/conf/bmvc/bmvc2017.html
[31] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 613–628.
[32] I. Nunes, M. B. Pereira, H. Oliveira, J. A. dos Santos, and M. Poggi, "Conditional reconstruction for open-set semantic segmentation," in *Proc. 2022 IEEE Int. Conf. Image Process.* (ICIP), 2022, pp. 946–950.
[33] J. Hong et al., "GOSS: Towards generalized open-set semantic segmentation," *Vis. Comput.*, vol. 40, no. 4, pp. 2391–2404, 2024.
[34] X. Li et al., "Pointflow: Flowing semantics through points for aerial image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 4215–4224.
[35] D. Pal, V. Bundele, R. Sharma, B. Banerjee, and Y. Jeppu, "Few-shot open-set recognition of hyperspectral images with outlier calibration network," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2022, pp. 2091–2100.
[36] Y. Sun, B. Liu, R. Wang, P. Zhang, and M. Dai, "Spectral–spatial MLP-like network with reciprocal points learning for open-set hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5513218.
[37] Z. Li, K. Bi, Y. Wang, Z. Fang, and J. Zhang, "Supervised contrastive learning for open-set hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, 2023, Art. no. 5509805.
[38] Z. Xie, P. Duan, W. Liu, X. Kang, X. Wei, and S. Li, "Feature consistency-based prototype network for open-set hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 9286–9296, Jul. 2024.
[39] D. Pal, S. Bose, B. Banerjee, and Y. Jeppu, "Extreme value meta-learning for few-shot open-set recognition of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5512516.
[40] B. Xi, W. Zhang, J. Li, R. Song, and Y. Li, "HyperCASR: Spectral-spatial open-set recognition with category-aware semantic reconstruction for hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 34, pp. 7642–7655, 2025.
[41] H. Sun et al., "Class-aware consistency learning for open-set semi-supervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, Art. no. 5535415.
[42] Q. Gui, H. Zhou, N. Guo, and B. Niu, "A survey of class-imbalanced semi-supervised learning," *Mach. Learn.*, vol. 113, no. 8, pp. 5057–5086, 2024.
[43] Q. Ma, J. Gao, B. Zhan, Y. Guo, J. Zhou, and Y. Wang, "Rethinking safe semi-supervised learning: Transferring the open-set problem to a close-set one," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 16324–16 333.
[44] Y. Fan, A. Kukleva, D. Dai, and B. Schiele, "SSB: Simple but strong baseline for boosting performance of open-set semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 16068–160 78.
[45] E. Wallin, L. Svensson, F. Kahl, and L. Hammarstrand, "Improving open-set semi-supervised learning with self-supervision," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2024, pp. 2356–2365.
[46] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," Technologies, vol. 9, no. 1, 2021, Art. no. 2. [Online]. Available: https://www.mdpi.com/2227-7080/9/1/2
[47] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 4, pp. 213–247, Dec. 2022.
[48] N. A. A. Braham, L. Mou, J. Chanussot, J. Mairal, and X. X. Zhu, "Self supervised learning for few shot hyperspectral image classification," in *Proc. IGARSS 2022 Int. Geosci. Remote Sens. Symp.*, 2022, pp. 267–270.
[49] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-invariance-covariance regularization for self-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=xm6YD62D1Ub
[50] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.* (PMLR), 2020, pp. 1597–1607.
[51] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9726–9735.
[52] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-supervised learning via redundancy reduction," in *Proc. 38th Int. Conf. Mach. Learn.*, Jul. 2021, pp. 12310–12320. [Online]. Available: https://proceedings.mlr.press/v139/zbontar21a.html
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
[54] A. Challa, S. Danda, B. D. Sagar, and L. Najman, "Watersheds for semi-supervised classification," *IEEE Signal Process. Lett.*, vol. 26, no. 5, pp. 720–724, May 2019.
[55] A. Challa, S. Danda, B. S. D. Sagar, and L. Najman, "Triplet-watershed for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5515014.
[56] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.
[57] R. O. Green et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, 1998. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425798000649
[58] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
[59] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15745–15 753.
[60] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[61] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[62] T. Lu, Y. Fang, W. Fu, K. Ding, and X. Kang, "Dual-stream class-adaptive network for semi-supervised hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5507511.
[63] B. Fu, Z. Cao, M. Long, and J. Wang, "Learning to detect open classes for universal domain adaptation," in *Proc. 16th Eur. Conf. Comput. Vis.–ECCV 2020*, 2020, pp. 567–583.
[64] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *Proc. 7th Python Sci. Conf.*., Pasadena, CA USA, 2008, pp. 11–15.



Matheus B. Pereira received the bachelor's degree in computer science from the Universidade Federal de Lavras, Lavras, Brazil, in 2018 and the master's degree in computer science from the Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2019. He is currently working toward the joint Ph.D. degree in computer science with the UFMG and Université Gustave Eiffel, Noisy-le-Grand, France.

He is currently an Assistant Professor with the Pontifical Catholic University of Minas Gerais, Belo Horizonte. His research interests include computer vision, pattern recognition, and machine and deep learning, primarily applied to remote sensing data analysis.



Jefersson A. dos Santos (Senior Member, IEEE) received the bachelor's degree in computer science from the University of Mato Grosso do Sul, Dourados, Brazil, in 2006, the master's degree in computer science from the University of Campinas (Unicamp), Campinas, Brazil, in 2009, and the joint Ph.D. degree in computer science from Unicamp and the University of Cergy-Pontoise, Cergy-Pontoise, France, in 2013

From 2013 to 2022, he was an Associate Professor with the Universidade Federal de Minas Gerais, Belo Horizonte, Brazil. He is currently an Assistant Professor (Lecturer) in computer vision with the University of Sheffield, Sheffield, U.K. He is the Founder of the Laboratory of Pattern Recognition and Earth Observation (PATREO) at the Department of Computer Science, UFMG, Brazil. His research interests include remote sensing image processing, computer vision, and machine learning..

Dr. Santos is an Associate Editor for IEEE GEOSCIENCE AND REMOTE SENSING LETTERSand Co-Chair for the ISPRS Working Group for AI/ML for Geospatial Data. He also serves as a PC Member for several refereed journals and conferences and is currently the Vice President of IEEE GRSS Brazilian Chapter. He held the prestigious CNPq Productivity Research Scholarship (2016–2022). In 2021, he was the recipient of a competitive research grant from the Serrapilheira Institute, Brazil.



Laurent Najman (Senior Member, IEEE) received the "Ingénieur" degree in engineering from the Ecole des Mines de Paris, Paris, France, in 1991, the Ph.D. degree [with the highest honor (Félicitations du Jury)] in applied mathematics from Paris-Dauphine University, Paris, in 1994, and the Habilitation á Diriger les Recherches degree in computer science from the University of Marne-la-Vallée, Paris, in 2006.

After earning his engineering degree, he was with the Central Research Laboratories of Thomson-CSF for three years, working on some problems of infrared image segmentation using mathematical morphology. He then joined a start-up company named Animation Science in 1995, as the Director of research and development. In 1998, he joined OCÉ Print Logic Technologies, as Senior Scientist. He worked there on various problems of image analysis dedicated to scanning and printing. In 2002, he joined the Computer Science Department of ESIEE Paris, where he is currently a Professor and a Member of the Laboratoire Informatique Gaspard Monge (LIGM), Université Gustave Eiffel, Noisy-le-Grand,- France. Since August 2025, he has been a Full Professor with the Department of Mathematics, Khalifa University, Abu Dhabi, UAE, where he is currently on leave from the LIGM. His current research interest include discrete mathematical morphology and discrete optimization.

Dr. Najman was the recipient of several awards, including the "European Information Technology Prize 1997" awarded by the European Commission (Esprit programme) and by the European Council for Applied Science and Engineering and the "Hottest Products of the Year 1996" awarded by the *Computer Graphics World* journal, for the technology of particle systems for computer graphics and scientific visualization, developed by the company under his technical leadership.

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - ROR identifier: 00x0ma614