



A Hierarchical Prototypical Network for Few-Shot Remote Sensing Scene Classification

Manal Hamzaoui^(✉), Laetitia Chapel, Minh-Tan Pham, and Sébastien Lefèvre

Université Bretagne-Sud, IRISA UMR 6074, 56000 Vannes, France
{`manal.hamzaoui,laetitia.chapel,minh-tan.pham,`
`sebastien.lefevre`}@univ-ubs.fr

Abstract. Few-shot learning (FSL) aims at making predictions based on a limited number of labeled samples. It is a hot topic in many fields such as natural language processing, computer vision and more recently, remote sensing. In this work, we focus on few-shot remote sensing scene classification which aims to recognize unseen scene categories at training stage from few or even a single labeled sample at test stage. Although considerable progress has been achieved in this topic, less attention has been paid to leveraging the prior structural knowledge. In this paper, we learn transferable visual features by introducing the class hierarchy which encodes the semantic relationship between the classes. We build on a prototypical network and we define *hierarchical prototypes* that allow us to encode the different levels of the hierarchy. Experiments conducted on the remote sensing NWPU-RESISC45 dataset demonstrate that the proposed hierarchical prototypical network acts as a regularizer and leads to better performance than the original network in the context of few-shot remote sensing scene classification.

Keywords: Few-shot learning · Class hierarchy · Scene classification · Remote sensing

1 Introduction

Scene classification is an important research topic in remote sensing which aims to automatically assign a specific semantic category to each remote sensing scene image. Deep learning frameworks have been applied to this problem and have achieved outstanding performance on most remote sensing image scene classification (RSISC) datasets. They essentially extract end-to-end features from images using deep neural networks such as Auto-Encoders [6] and Convolutional Neural Networks (CNN) [14]. However, most of supervised deep remote sensing scene classification algorithms are “data-hungry” as they require a large amount of labeled data for training. When the labeled data are insufficient, there would be an obvious over-fitting and irrelevant extracted features, leading to a degradation of classification performance. However, obtaining labeled samples may be tough as it is labor-intensive, time-consuming and may need strong human expertise.

Inspired by the human ability to learn new abstract concepts from very few, or even one, examples and to generalize quickly to new instances [15], few-shot learning (FSL) was introduced as one of the alternative ways to deal with the “data-hungry” issue. FSL methods can be divided into three categories [18]: metric learning, meta-learning and transfer learning. Metric learning methods learn a distance function that brings samples from the same category as close as possible in the feature space while pushing samples from other categories as far away as possible. As for meta-learning, also known as learning to learn, it is the most common approach in FSL, which efficiently optimizes the model parameters to new tasks. Transfer learning aims at using the knowledge gained from relevant tasks towards new tasks, *e.g.* fine-tuning the pre-trained models is a powerful transfer method.

Recently, the combination of meta-learning and metric learning has been one of the most studied approaches in FSL for natural image classification [17, 20] and for remote sensing scene classification [22]. First, based on meta-learning, these approaches construct tasks with few labeled samples, which enhances the generalization performance of the model for new tasks. Then, the similarity between image features is measured to make predictions. Some of related methods include relation network [19], classical matching network [20] and prototypical network [17].

In addition to the sparsity of labeled data, it may be more challenging to classify remote sensing data than natural images. Indeed, remote sensing scene images may present confusing visual similarity between different classes. The large intra-class variation may even exceed the inter-class variance, thus similar semantic classes may present significant visual dissimilarity [5]. Another challenge is that remote sensing images are top-down views and contain inevitably many objects that are not relevant to the semantic class of the scene [10]. Yet, this characteristic could be very useful in multi-label or hierarchical classification tasks where several levels of semantic granularity are considered.

In the recent years, many approaches were proposed to tackle the problem of few-shot remote sensing scene classification (FSRSSC). In [9], the authors adopted the attention mechanism to delve into the inter-channel and inter-spatial relationships to discover discriminative regions in the remote sensing scene images. The authors in [3] used a Siamese-prototype network with prototype self-calibration and inter-calibration to learn more discriminative prototypes. In [22], the authors introduced a pre-training step on the base data to provide better initialization of the feature extractor and performed the few-shot remote sensing scene classification using cosine distance metric. However, to the best of our knowledge, the majority of these methods have focused only on visual scene information to improve feature representations without considering semantic knowledge that may exist within these classes. Yet this type of semantic knowledge about classes, which can consist of attributes, word embeddings or even a knowledge graph (*e.g.* WordNet [13]), is commonly used in zero-shot learning (ZSL) and increasingly in few-shot natural images classification approaches.

Semantic knowledge is not a novelty in ZSL, since this task can not be accomplished without such knowledge. However, in FSL, this semantic knowledge has hardly been used until recently. [2] proposed the TriNet to tackle the “1-shot”

task by synthesizing the instance features from the semantic space which is given by the label embeddings. In [21], the authors proposed a method called Semantic Guided Attention (SEGA) mechanism which leverages semantic knowledge to guide the visual perception in learning the discriminative visual features of each class. Most of these FSL approaches that introduce semantic knowledge involve the text modality. However, few attention has been paid to knowledge transfer based on the class hierarchy which is either built using text modality as in [8] or already predefined as in [11]. In [8], the authors proposed a hierarchical image recognition approach by performing Softmax optimization on all levels of the class hierarchy. This allows learning transferable visual features through this class hierarchy which encodes semantic relationships between seen and unseen classes. In [11], a class hierarchy was introduced to address the multi-class FSL problem. The authors proposed a “memory-augmented hierarchical-classification network (MahiNet)” model which leverages the hierarchy as prior knowledge to train a coarse-to-fine classifier where each coarse class can cover multiple finer classes.

According to [11], FSL with knowledge transfer can be accomplished independently of an additional modality such as text and yields competitive performances, when the class hierarchy is known or easily obtained, which fits well with our research interests. This class hierarchy has been successfully applied to traditional supervised learning tasks. It can be introduced in the learning process according to three approaches [1]: a label-embeddings approach, a hierarchical loss or hierarchical architectures. In label-embeddings approach, a mapping function is used to encode class relationship information and associate it to class representations such as *soft-labels* [1]. Hierarchical loss-based methods adjust the loss to be optimized by assigning a higher penalty to predictions that are distant from the true label in the class hierarchy, such as *hierarchical cross-entropy loss* [1]. As for hierarchical networks, they introduce the class hierarchy into the classifier architecture without necessarily changing the loss function, allowing them to make super-class predictions at early layers and fine predictions at later layers.

The remote sensing classes can be easily arranged in a hierarchical structure following well-known organizations such as Corine Land Cover (CLC), the European Nature Information System (EUNIS) habitat classification scheme or other structures such as done in [12] where they propose a hierarchical organization of the scene classes of the PatternNet [25] remote sensing scene dataset.

In this work, we rely on the semantic knowledge associated with scene classes through their hierarchical organization. We build on prototypical networks to define a hierarchical variant: in a nutshell, hierarchical prototypes are attached to each level of the hierarchy, allowing us to first consider high-level aggregated information before making a fine prediction. We show on a remote sensing dataset that it acts as a regularizer, giving better performances not only at the top nodes of the hierarchy, but also at the leaf classes. We also show that it performs better than *soft-labels* [1] that we introduce for the first time (to our knowledge) in a remote sensing few-shot learning context.

The remainder of our paper is organized as follows. In Sect. 2, we provide some details about FSL and prototypical networks. Section 3 presents in depth the proposed method. We describe the experimental setup and the obtained results in Sect. 4. Conclusion and future works are given in Sect. 5.

2 Few-Shot Classification with Prototype Learning

2.1 Problem Formulation of the FSL

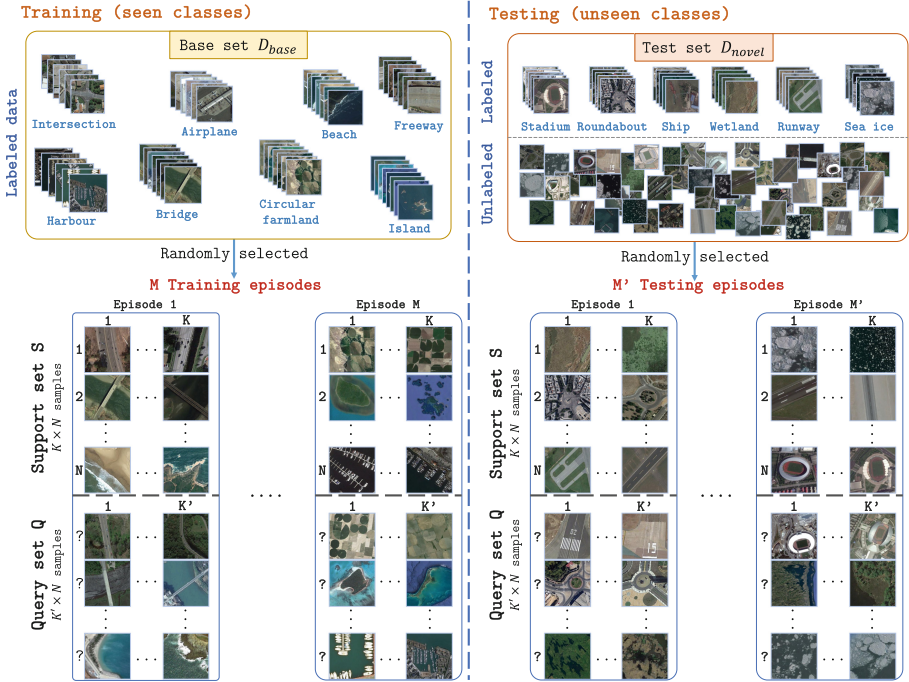


Fig. 1. Illustration of N-way K-shot classification episodes. The left side shows the M episodes of the training step; each episode consists of $N \times K$ support samples and $N \times K'$ query samples. The testing step is similarly defined on M' episodes, as shown on the right.

In few-shot classification, we assume that we have two sets, a large labeled training set, referred to as the base set D_{base} , and a test set with few labeled images per class, the novel set D_{novel} . The classes that constitute the base and novel sets, denoted C_{base} and C_{novel} respectively, are disjoint $C_{base} \cap C_{novel} = \emptyset$. To mimic the sparsity of the test data in the training stage, we adopt the N-way K-shot strategy (an episodic learning strategy) used in various FSL studies [17, 20], in which N refers to the number of classes and K (usually set to 1 or 5)

is the number of labeled images per classes during a training/testing episode. For each episode, we randomly sample a subset of N classes out of C_{base} during training and out of C_{novel} during testing, which we denote C_e , to construct the episode support set S and the episode query set Q . During a training episode, we randomly sample K labeled images from D_{base} for each class $c \in C_e$, resulting in the episode support set $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$, where x_i is an image and $y_i \in C_e$ its corresponding label. Similarly for the episode query set Q , K' labeled images are sampled from the base set D_{base} for each class $c \in C_e$, resulting in $Q = \{(x_q, y_q)\}_{q=1}^{N \times K'}$. In this training step, the support set S and the query set Q are used to learn the model that projects the input images into the feature space. The testing step is also carried out with the same episodic strategy where we have an unlabeled query set Q (drawn from D_{novel}) for which we want to predict the class label of each query sample $x_q \in Q$ using the labeled support set S (also drawn from D_{novel}). Figure 1 shows a visualization of the N-way K-shot episodes.

2.2 Prototypical Networks

Prototypical networks [17] adopt an episodic strategy to train a meta-learner classifier \mathcal{M} . Given an episode with a support set S and a query set Q , we compute the representations of the images in S using the meta-learner feature extractor f_Φ (a neural network such as CNN) parameterized by Φ . Thereafter, the representations are averaged to compute the prototypes p^c for each class $c \in C_e$ as follows:

$$p^c = \frac{1}{K} \sum_{(x_i, y_i) \in S^c} f_\Phi(x_i) \quad (1)$$

where S^c is the subset of the episode support set S that contains the samples of class $c \in C_e$.

To optimize the feature extractor f_Φ , we minimize the loss function:

$$\mathcal{L} = -\frac{1}{N \times K'} \sum_{c \in C_e} \sum_{(x_q, y_q) \in Q^c} \log p_\Phi(y_q = c | x_q) \quad (2)$$

where Q^c is the subset of the episode query set Q that contains the samples of class c , $p_\Phi(y_q = c | x_q)$ is the probability of predicting a query sample $(x_q, y_q) \in Q$ as class c and is given as:

$$p_\Phi(y_q = c | x_q) = \frac{\exp(-d(f_\Phi(x_q), p^c))}{\sum_{c' \in C_e} \exp(-d(f_\Phi(x_q), p^{c'}))} \quad (3)$$

where $d(\cdot)$ represents a certain distance measurement, such as the Euclidean distance [17] or the Cosine distance [22].

3 A Hierarchical Prototypical Network for Few-Shot Image Classification

3.1 Overall Framework

We propose a meta-learning framework whose complete pipeline is illustrated in Fig. 2 to solve the few-shot problem when a hierarchy that describes the organization between the classes is available. We train a meta-learner classifier \mathcal{M} by adopting an episodic training strategy. During training stage, using the support set S , we compute N prototypes $\mathcal{P} = \{p^c\}_{c \in C_e}$ for each class in the current task (episode) and N_h hierarchical prototypes for their super-classes. The query features are then compared to both the scene and the hierarchical prototypes, allowing us to compute an episodic error at different levels of the class hierarchy \mathcal{H} to be minimized and used to finetune the parameters Φ of the feature extractor f_Φ . At testing stage, the parameters Φ of the feature extractor f_Φ are fixed and the meta-learner classifier \mathcal{M} is evaluated on a set of episodes sampled from the novel classes in D_{novel} .

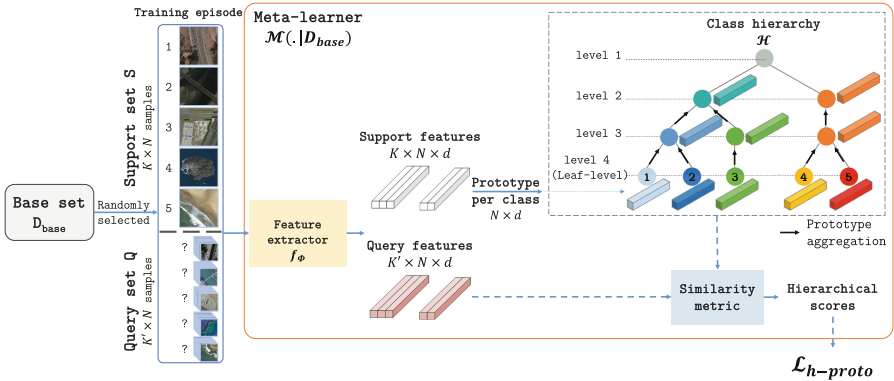


Fig. 2. Overall framework of the proposed hierarchical prototypical network for few-shot image classification. In this example (one-shot), $N = 5$, $N_h = 5$, $K = 1$, $K' > 1$ (usually set to 15).

3.2 Hierarchical Prototypical Network

Some works have already attempted to introduce the class hierarchy knowledge into the few-shot classification process. In [8], the authors suggested to perform a Softmax optimization over the different levels of the class hierarchy to enable knowledge transfer from seen to unseen classes. Here, we rather rely on the prototypical networks and introduce the hierarchy knowledge thanks to the definition of hierarchical prototypes. The overall idea is to regularize the latent space by putting closer classes that are in the same branch of the class hierarchy, and pushing apart classes that have common ancestors in higher levels of the class hierarchy.

To properly formulate our approach, given an episode, we first compute the prototypes per class which are prototypes at the leaf-level of the class hierarchy \mathcal{H} (following Eq. 1). We then compute the hierarchical prototypes by aggregating the leaf-level prototypes according to \mathcal{H} . The prototypes of the super-classes $c \in C_e^l$ (the hierarchical prototypes) at level l ($1 < l < L$ with $l = 1$ the root node and $L = \text{height}(\mathcal{H})$) are denoted as $\mathcal{P}_l = \{p_l^c\}_{c \in C_e^l}$ and computed as the mean of support samples of the super-class sub-tree S_l^c similarly to Eq. 1:

$$p_l^c = \frac{1}{|S_l^c|} \sum_{(x_i, y_i) \in S_l^c} f_{\Phi}(x_i) \quad (4)$$

Note that when $l = L$, the prototypes at level l are the prototypes at the lowest level of \mathcal{H} (leaf-level prototypes).

The hierarchical prototypical network outputs a distribution over classes for each query sample $x_q \in Q$ at different levels of \mathcal{H} , based on a Softmax over the distances to the prototypes of each level l in \mathcal{H} . We then formulate the probability of predicting the query features $f_{\Phi}(x_q)$ and the prototype p_l^c of its super-class c at level l in \mathcal{H} as formulated in Eq. 3 as:

$$p_{\Phi}(y_q^l = c | x_q) = \frac{\exp(-d(f_{\Phi}(x_q), p_l^c))}{\sum_{c' \in C_e^l} \exp(-d(f_{\Phi}(x_q), p_l^{c'}))}, \quad (5)$$

where y_q^l is the ancestor of y_q at level l , C_e^l represents the super-classes at level l at the current episode.

We therefore optimize a new loss function given as

$$\mathcal{L}_{\mathcal{H}\text{-proto}} = \sum_{l=2}^L \lambda_l \mathcal{L}_l \quad (6)$$

where $\lambda_l = \frac{\gamma^{l-1}}{\sum_{l'=2}^L \gamma^{l'-1}}$, γ is a hyper-parameter that controls the importance of each level in the hierarchy and $\sum_{l=2}^L \lambda_l = 1$. \mathcal{L}_l represents the prototypical network loss at level l of the class hierarchy \mathcal{H} .

As such, we can tune the importance of each level of the hierarchy into the learning process: by choosing low values of γ , we put more importance into organizing the higher levels of the hierarchy; a value close to one gives the same importance for all the levels; a high value tends to behave like the *flat* cross entropy loss formulation.

4 Few-Shot Learning for Remote Sensing Scene Classification

We evaluate the performance of our hierarchical prototypical approach in a few-shot scene classification task. We first present the remote sensing scene dataset we consider in our study, since its labels are hierarchically organized. Then, we

describe the parameters and hyper-parameters setting. Finally, in order to assess the interest of introducing semantic knowledge into the few-shot scene classification task, we compare the classification results of the proposed approach to some baseline methods for the 5-way 5-shot and 5-way 1-shot tasks.

4.1 Dataset Description

The NWPU-RESISC45 [4] dataset is a widely used benchmark for remote sensing image scene classification. It consists of 31 500 images of 256×256 pixels; the spatial resolution varies from approximately 30 to 0.2 m per pixel. It covers 45 scene categories, each with 700 RGB images, which can be organized hierarchically. Following [12] in which the authors propose a hierarchical organization of the scene classes of the PatternNet [25] remote sensing scene dataset, we construct a tree-like arrangement of these scene classes which reflects their semantic relationships. We note that the leaf level of the constructed class hierarchy corresponds to the original scene classes of the dataset. A sub-tree of the 3-level label tree is shown in Fig. 3.

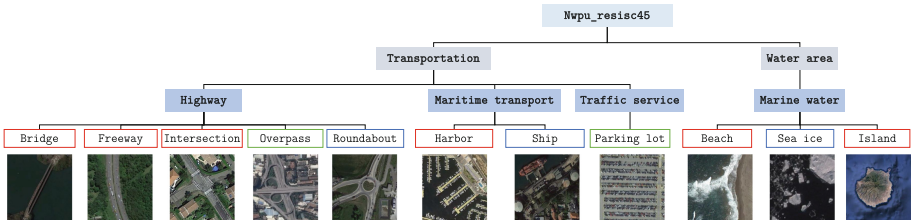


Fig. 3. Sub-tree of the proposed label tree for the NWPU-RESISC45 remote sensing dataset. The leaves correspond to the classes, the distance between two given classes is the height of the subtree at the Lowest Common Ancestor (LCA) node, and can take one of the following values: 0, 1, 2, and 3. The meta-train, meta-validation, and meta-test categories are leaves with red, green, and blue boxes respectively. (Color figure online)

For a fair comparison, we adopt the same split as done in [22]. We split the dataset into three disjoint subsets: meta-training D_{base} , meta-validation D_{val} , and meta-test D_{novel} containing 25, 8, and 12 categories, respectively. We note that the meta-validation set is used for hyper-parameter selection in the meta-training step. The meta-training set is further divided into three subsets: training, validation, and test sets. In our experiments, we follow [22] and resize all the images to 80×80 pixels to fit our designed feature extractor.

4.2 Implementation Details

Following recent FSRSSC studies [10, 22–24], we utilize ResNet-12 as a backbone for feature extraction. We also adopt the pre-training strategy as suggested in [22] to better initialize the meta-learner feature extractor.

We train our meta-learner for 1000 epochs with an early stopping of 50 epochs. The best model parameters were obtained within the first 300 epochs. In standard deep learning, an epoch implies that the entire train set passes through the deep neural network once. However, in meta-learning, an epoch is a set of episodes randomly sampled from the base set D_{base} , which we set to 1000 episodes per epoch. We optimize the model based on the average loss of 4 episodes, i.e. the batch size is set to 4 episodes. We use SGD optimizer to update the network parameters with a momentum set to 0.9 and a weight decay set to 0.0005. The learning rate is fixed to 0.001. After each training epoch, we test our model on a validation set D_{val} by randomly sampling 1000 episodes, the network weights with the lowest validation loss are retained as the best results. For the hyper-parameter γ , we assigned different values ($\gamma = 1$, $\gamma < 1$ and $\gamma > 1$) in order to observe its impact on the framework performances.

For the meta-testing stage, we conduct a 5-way 1-shot and 5-way 5-shot classification following the widely used meta-learning protocol. We evaluate the best model on 2000 randomly sampled episodes from the test set D_{novel} . Following the FSL evaluation protocol [17], for 5-way K-shot episode, we randomly sample 5 classes from the unseen classes C_{novel} , K images per class to form the support set S , and 15 images per class to form the query set Q , making a total of $5 \times (K + 15)$ images per episode.

4.3 Evaluation Metrics

We use two metrics to evaluate the performance of the different methods:

- The classification accuracy, computed at different levels of the class hierarchy;
- The hierarchical precision [16] which is defined as the total number of common ancestors between the predicted class and the true class divided by the total number of ancestors of the predicted classes:

$$hp = \frac{\sum_i |\hat{Y}_i \cap Y_i|}{|\sum_i \hat{Y}_i|} \quad (7)$$

where $\hat{Y}_i = \{\hat{y}_i \cup Ancestor(\hat{y}_i, \mathcal{H})\}$ is the set consisting of the most specific predicted class for test example i and all its ancestor classes in \mathcal{H} except the root node and $Y_i = \{y_i \cup Ancestor(y_i, \mathcal{H})\}$ is the set consisting of the most specific true class for test example i and all its ancestor classes in \mathcal{H} except the root node.

For all evaluation metrics, we report the average of the test episodes with a 95% confidence interval.

4.4 Experimental Results

In both 5-way K-shot configurations, $K = 1$ or 5, we compare our method to the original *flat* prototypical network [17] and to the approach proposed in [22]

that uses the cosine metric as a similarity function, which we denote by *ProtoNet* and *c-ProtoNet* respectively. We re-implement both methods with their related parameter setting according to [22] and use ResNet-12 as a backbone for a fair comparison. We also compare our prototypes (namely *h-ProtoNet*) with the results yielded by the soft-label method [1], which allows taking into account the class hierarchy within the learning process. Results are provided in Table 1 for $K = 5$ and Table 2 for $K = 1$.

Table 1. 5-shot classification results computed on the test set of the NWPU-RESISC45 dataset at different levels of the class hierarchy: overall acc represents the classification accuracy at the leaves (level 4) and thus the NWPU-RESISC45 classes; L3-acc and L2-acc give the accuracy at level 3 and level 2, respectively; hp is the hierarchical precision. All accuracy results are averaged over 2000 test episodes and are reported with a 95% confidence interval.

Method	hyp-param	Overall acc	L3-acc	L2-acc	hp
ProtoNet [17]	/	83.76 \pm 0.13	84.80 \pm 0.04	85.62 \pm 0.09	84.72 \pm 0.08
c-ProtoNet [22]	10	80.21 \pm 0.59	82.11 \pm 1.80	84.81 \pm 4.08	82.38 \pm 2.16
Soft-labels [1]	4	84.22 \pm 0.25	85.35 \pm 0.23	86.19 \pm 0.23	85.25 \pm 0.23
h-ProtoNet (ours)	0.5	84.90 \pm 0.25	86.01 \pm 0.22	86.72 \pm 0.23	85.88 \pm 0.22
h-ProtoNet (ours)	1	85.11 \pm 0.23	86.10 \pm 0.22	86.81 \pm 0.22	86.01 \pm 0.21
h-ProtoNet (ours)	2	84.95 \pm 0.23	85.93 \pm 0.22	86.65 \pm 0.20	85.85 \pm 0.22

For both 1-shot and 5-shot cases, our proposed *h-ProtoNet* achieves the highest accuracy and outperforms both flat prototypes and the soft-labels hierarchical loss. We obtain the best performance with $\gamma = 1$, that is to say when all the (hierarchical) prototypes have the same weights. When we put more weights on the prototypes that correspond to the higher level of the hierarchy (corresponding to nodes close to the root, $\gamma < 1$) or to those that correspond to the leaves ($\gamma > 1$), we obtain degraded performances, that are still better than the other methods. Note that this value of $\gamma = 1$ would have been selected if we perform a cross-validation on the validation set. We argue that the improvement observed in the case of the hierarchical prototypes is due to an efficient regularization of the latent space, with a loss that encourages leaves within the same branch of the level hierarchy to be closer. As such, the performances at level 2 and 3 are improved, but also the overall accuracy.

Table 2. 1-shot classification results computed on the test set of the NWPU-RESISC45 dataset at different levels of the class hierarchy: overall acc represents the classification accuracy at the leaves (level 4) and thus the NWPU-RESISC45 classes; L3-acc and L2-acc give the accuracy at level 3 and level 2, respectively; hp is the hierarchical precision. All accuracy results are averaged over 2000 test episodes and are reported with a 95% confidence interval.

Method	hyp-param	Overall acc	L3-acc	L2-acc	hp
ProtoNet [17]	/	65.67 ± 0.46	67.39 ± 0.45	69.49 ± 0.45	67.52 ± 0.44
c-ProtoNet [22]	10	65.64 ± 0.45	67.52 ± 0.45	69.62 ± 0.45	67.59 ± 0.44
Soft-labels [1]	3	65.65 ± 0.45	67.24 ± 0.45	69.56 ± 0.45	65.65 ± 0.45
h-ProtoNet (ours)	0.5	67.23 ± 0.45	69.02 ± 0.45	71.41 ± 0.45	69.22 ± 0.43
h-ProtoNet (ours)	1	67.25 ± 0.45	68.94 ± 0.45	71.23 ± 0.45	69.14 ± 0.43
h-ProtoNet (ours)	2	66.86 ± 0.45	68.74 ± 0.45	70.92 ± 0.45	68.84 ± 0.44

5 Conclusion and Future Works

Few-shot learning has captured the attention of the remote sensing community thanks to the great success it has achieved in other fields. In many cases, when dealing with the problem of scene classification, the organization of the classes is defined in a hierarchical manner, with classes being semantically closer than some others. In this work, we present a novel prototypical network which defines hierarchical prototypes that match the nodes of the label hierarchy¹. We evaluate our method on a benchmarked [4] remote sensing scene dataset in a few-shot learning context and we show that hierarchical prototypes ensure a regularization of the latent space, providing higher performance than *flat* prototypes but also than a competitive hierarchical loss introduced in another context.

In future work, we plan to investigate the use of the hierarchical prototypes on other tasks such as semantic segmentation, where we consider adding spatial information among regions to output meaningful hierarchical prototypes. We also intend to use graph prototypical networks instead of prototypical networks, which better take into account the class relationships. We further consider relying on other metric spaces than the Euclidean one, *e.g.* hyperbolic spaces that are known to better encode the distances when the data are hierarchically-organized. Thus, a follow-up of this work could be to build hyperbolic prototypes to enforce the hierarchical information into the learning process.

Acknowledgement. This work was supported by the ANR Multiscale project under the reference ANR-18-CE23-0022.

¹ We recently became aware of a paper that proposes a similar approach to classify audio data in the FSL context [7]. The difference lies rather in the experimental part in which we use a deeper network and a pre-training step.

References

1. Bertinetto, L., Müller, R., Tertikas, K., Samangoeei, S., Lord, N.A.: Making better mistakes: leveraging class hierarchies with deep networks. In: CVPR, pp. 12503–12512. Computer Vision Foundation/IEEE (2020)
2. Chen, Z., et al.: Multi-level semantic feature augmentation for one-shot learning. *IEEE Trans. Image Process.* **28**(9), 4594–4605 (2019)
3. Cheng, G., et al.: SPNet: siamese-prototype network for few-shot remote sensing image scene classification. *IEEE Trans. Geosci. Remote. Sens.* **60**, 1–11 (2022)
4. Cheng, G., Han, J., Lu, X.: Remote sensing image scene classification: benchmark and state of the art. *Proc. IEEE* **105**(10), 1865–1883 (2017)
5. Cheng, G., Yang, C., Yao, X., Guo, L., Han, J.: When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **56**(5), 2811–2821 (2018)
6. Esam, O., Yakoub, B., Naif, A., Haikel, A., Farid, M.: Using convolutional features and a sparse autoencoder for land-use scene classification. *Int. J. Remote Sens.* **37**(10), 2149–2167 (2016)
7. Garcia, H.F., Aguilar, A., Manilow, E., Pardo, B.: Leveraging hierarchical structures for few-shot musical instrument recognition. In: ISMIR, pp. 220–228 (2021)
8. Li, A., Luo, T., Lu, Z., Xiang, T., Wang, L.: Large-scale few-shot learning: knowledge transfer with class hierarchy. In: CVPR, pp. 7212–7220. Computer Vision Foundation/IEEE (2019)
9. Li, L., Han, J., Yao, X., Cheng, G., Guo, L.: DLA-MatchNet for few-shot remote sensing image scene classification. *IEEE Trans. Geosci. Remote. Sens.* **59**(9), 7844–7853 (2021)
10. Li, X., Li, H., Yu, R., Wang, F.: Few-shot scene classification with attention mechanism in remote sensing. *J. Phys. Conf. Ser.* **1961**, 012015 (2021)
11. Liu, L., Zhou, T., Long, G., Jiang, J., Zhang, C.: Many-class few-shot learning on multi-granularity class hierarchy. *CoRR abs/2006.15479* (2020)
12. Liu, Y., Liu, Y., Chen, C., Ding, L.: Remote-sensing image retrieval with tree-triplet-classification networks. *Neurocomputing* **405**, 48–61 (2020)
13. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
14. Nogueira, K., Penatti, O.A.B., dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **61**, 539–556 (2017)
15. Shi, X., Salewski, L., Schiegg, M., Welling, M.: Relational generalized few-shot learning. In: BMVC. BMVA Press (2020)
16. Silla, C.N., Jr., Freitas, A.A.: A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.* **22**(1–2), 31–72 (2011)
17. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NIPS, pp. 4077–4087 (2017)
18. Sun, X., et al.: Research progress on few-shot learning for remote sensing image interpretation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 2387–2402 (2021)
19. Sung, F., et al.: Learning to compare: relation network for few-shot learning. In: CVPR, pp. 1199–1208. Computer Vision Foundation/IEEE Computer Society (2018)
20. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: NIPS, pp. 3630–3638 (2016)

21. Yang, F., Wang, R., Chen, X.: SEGA: semantic guided attention on visual prototype for few-shot learning. CoRR abs/2111.04316 (2021)
22. Zhang, P., Bai, Y., Wang, D., Bai, B., Li, Y.: Few-shot classification of aerial scene images via meta-learning. *Remote Sens.* **13**(1), 108 (2021)
23. Zhang, P., Fan, G., Wu, C., Wang, D., Li, Y.: Task-adaptive embedding learning with dynamic kernel fusion for few-shot remote sensing scene classification. *Remote Sens.* **13**(21), 4200 (2021)
24. Zhang, P., Li, Y., Wang, D., Wang, J.: RS-SSKD: self-supervision equipped with knowledge distillation for few-shot remote sensing scene classification. *Sensors* **21**(5), 1566 (2021)
25. Zhou, W., Newsam, S.D., Li, C., Shao, Z.: PatternNet: a benchmark dataset for performance evaluation of remote sensing image retrieval. CoRR abs/1706.03424 (2017)