remote sensing

MDPI

Article

Meta-FSEO: A Meta-Learning Fast Adaptation with Self-Supervised Embedding Optimization for Few-Shot Remote Sensing Scene Classification

Yong Li [1], Zhenfeng Shao [1,*], Xiao Huang [2], Bowen Cai [3] and Song Peng [1]

[1] State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; yong.li@whu.edu.cn (Y.L.); songpeng@whu.edu.cn (S.P.)
[2] Department of Geosciences, University of Arkansas, Fayetteville, AR 72701, USA; xh010@uark.edu
[3] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; caibowen@whu.edu.cn
* Correspondence: shaozhenfeng@whu.edu.cn

Abstract: The performance of deep learning is heavily influenced by the size of the learning samples, whose labeling process is time consuming and laborious. Deep learning algorithms typically assume that the training and prediction data are independent and uniformly distributed, which is rarely the case given the attributes and properties of different data sources. In remote sensing images, representations of urban land surfaces can vary across regions and by season, demanding rapid generalization of these surfaces in remote sensing data. In this study, we propose Meta-FSEO, a novel model for improving the performance of few-shot remote sensing scene classification in varying urban scenes. The proposed Meta-FSEO model deploys self-supervised embedding optimization for adaptive generalization in new tasks such as classifying features in new urban regions that have never been encountered during the training phase, thus balancing the requirements for feature classification tasks between multiple images collected at different times and places. We also created a loss function by weighting the contrast losses and cross-entropy losses. The proposed Meta-FSEO demonstrates a great generalization capability in remote sensing scene classification among different cities. In a five-way one-shot classification experiment with the Sentinel-1/2 Multi-Spectral (SEN12MS) dataset, the accuracy reached 63.08%. In a five-way five-shot experiment on the same dataset, the accuracy reached 74.29%. These results indicated that the proposed Meta-FSEO model outperformed both the transfer learning-based algorithm and two popular meta-learning-based methods, i.e., MAML and Meta-SGD.

Keywords: meta-learning; few-shot learning; remote sensing scene classification; self-supervised; transfer learning

Citation: Li, Y.; Shao, Z.; Huang, X.; Cai, B.; Peng, S. Meta-FSEO: A Meta-Learning Fast Adaptation with Self-Supervised Embedding Optimization for Few-Shot Remote Sensing Scene Classification. *Remote Sens.* **2021**, *13*, 2776. https://doi.org/10.3390/rs13142776

Received: 25 May 2021
Accepted: 7 July 2021
Published: 14 July 2021

1. Introduction

The advancement of remote sensing image acquisition technology and the popularization of high-performance computing has promoted the utilization of an increasing amount of remote sensing data and computing resources. Deep learning algorithms have received increased interest in the field of remote sensing image processing in recent years [1–3]. Deep learning algorithms effectively extract features in the most common end-to-end methods through deep neural nets such as auto-encoders, the profound belief network, and convolutional neural networks [4–6]. The performance of deep learning, however, is heavily influenced by the size of the learning sample, and deep learning algorithms often assume that training and prediction data are independent and uniformly distributed, which is rarely the case given the attributes and properties of different data sources. Thus, these approaches require a substantial amount of data labeled for training.

However, labeling remote sensing images is a time-consuming and arduous process, and high-resolution remote sensing images are difficult to obtain in specific urban locations. Due to data limitations, a limited number of remote sensing images can be obtained in specific cities due to geographical limitations and weather restrictions. In addition, the same ground object in different areas can appear quite different, posing challenges to traditional deep learning techniques when learning samples are scarce. The performance of data-driven forecasting models in practical applications is greatly restricted by this deficiency.

In certain cases, the priori information of remote sensing images of known cities can be used to comprehensively construct a city feature extraction model with strong generalization ability using just a few samples. Such a learning schema allows models to quickly generalize from one study area to another even when labeled data are insufficient to support the training of traditional deep learning models. This schema leads to improved model performance [7]. Transfer learning extends this concept and has been widely adopted. For example, Li et al. [8] proposed a deep convolutional neural network model-based classification method and a transfer learning method to decrease overfitting problems and increase the accuracy of classification with limited labeled data. To investigate land cover SAR images from all over the world, Huang et al. [9] applied an extremely deep residual network with a transitive transfer learning method, extracting information from natural photos and transferring it to remotely sensed images and then to SAR data. By evaluating how the specialization of the CNN model impacts the transfer process, Pires et al. [10] found that the performance of transfer learning from models trained on natural images with larger data exceeds the performance of training models directly on remote sensing images with few data. However, transfer learning usually requires pre-trained weights derived from a large number of datasets, which are further fine-tuned towards the target dataset. Such a fine-tuning process is computationally demanding. If few labeled samples exist, existing transfer learning models usually fail to learn the new data distribution, leading to overfitting. Therefore, the transfer learning models cannot achieve satisfactory performance in few-shot scenarios.

We believe that our data model should mimic human thought, i.e., learning from a few instances and adjusting quickly as more data are obtained. However, this type of fast and flexible learning is challenging because the algorithm must combine previous experiences with a few numbers of newly added information while avoiding over-fitting the new data [11]. All these problems encourage us to examine the problem of the few-shot learning [12,13]. Solving the few-shot problem in the remote sensing domain alleviates the necessity to collect a large number of labeled training samples, which is usually a cumbersome process. The recent successful application of meta-learning in classification, regression, and reinforcement learning has established a new venue where few-shot problems can be solved [14]. Meta-learning and deep learning are different in training methods. In order to simulate the problem of few-shot learning, meta-learning learns from a range of tasks while deep learning learns from a set of data; each task contains a labeled training set and a labeled testing set [15,16].

In remote sensing images, urban land surfaces in different regions under different seasons can be very different, demanding rapid generalization of few-shot remote sensing data between different urban regions. Some researchers have made some related progress. For example, Li et al. [17] proposed a framework called RS-MetaNet that learns a metric module that can achieve high performance in few-shot remote sensing scene classification through a series of tasks. Ruswurm et al. [18] demonstrated that remote sensing tasks across geographies could be restructured as one meta-learning problem, despite them only investigating a few existing models. In order to explore a potential solution to these issues, we proposed Meta-FSEO, a novel model for improving the performance of few-shot remote sensing scene classification in varied urban scenarios. The proposed algorithm was designed to learn and adapt quickly to different geographical regions from just

a few samples. A meta-learning model should adapt and generalize to execute new tasks and new urban areas never experienced in the training process.

This paper contributes to the literature in three major aspects:

(1) We proposed a meta-learning algorithm called Meta-FSEO to improve the generalization performance of classification models in multiple urban conditions under a few-shot scenario. The proposed Meta-FSEO allows quick generalization to the data from unknown cities by training on the data of known cities according to task-level samples.
(2) We designed a self-supervised comparison module that effectively balances the requirements for feature classification tasks between multiple images collected at different times and places.
(3) We designed a loss function that combines the contrast loss and the cross-entropy loss weighting, aiming to achieve high-accuracy generalization capabilities.

2. Materials and Methods

Numerous attempts have been made to realize few-shot remote sensing scene classification, and most adopt a small amount of data from known city data to quickly generalize to the data from other unknown cities. In this section, we introduce relevant work on transfer learning and meta-learning, followed by a detailed discussion of the proposed Meta-FSEO model.

2.1. Transfer Learning

Transfer learning is a process of transferring knowledge from the source domain/task, where training data are plentiful, to a target domain/task, where training data are sparse [19]. Domain adaptation is a unique form of transfer learning with the same source/target activities but different source/target domains. Fine-tuning, aiming to adapt the pre-trained model to new tasks, is an efficient transfer method for deep learning models [20]. Saikia et al. [21] have shown that competitive performance can be reached using a strong hyperparameter optimization method applied on a carefully designed validation metric appropriate for few-shot learning. Chen et al. [22] employed transfer learning to construct an end-to-end trainable aircraft detection model by adopting a single deep convolutional neural network with a limited training sample. Li et al. [23] proposed a heterogeneous transfer learning framework to build a shared source and target data space and a new iterative weighting technique for weighing the source samples.

2.2. Self-Supervised Transformers

A self-supervised transformer [24] uses a transformer as the backbone network and achieves prediction through data transformation and comparison learning. Transformers were originally used in machine translation applications and later became a mainstream backbone in the neuro-linguistic programming (NLP) field [25]. They have now become a standard tool such as the generative pre-training (GPT) [26] and bidirectional encoder representations from transformers (BERT) models [27] blocks in neuro-linguistic programming. The long-range, self-attention behavior makes transformers an effective tool to tackle the non-local, relational nature of languages. Transformers are generally applied in the field of computer vision. At present, there are two main ways to implement transformers. One way is to incorporate them into the backbone. The recent work on visual transformers (ViT) [28] has explored this possibility. The other way is to combine transformers with CNN networks. For example, Botnet [29] greatly improved baselines on instance segmentation and object detection by replacing convolution layers with transformer modules in the final three bottleneck blocks of a ResNet. Our meta-learning approach for optimizing the inner loop of the query set was inspired by self-supervised transformers. Given the fact that transformers usually fail to extract features in a robust

manner, we use the transformers to quickly optimize features extracted by the CNN network.

2.3. Meta-Learning Background Knowledge

Meta-learning, often known as "learning to learn", refers to the process of enhancing a learning algorithm through numerous learning episodes [14]. Few-shot learning tries to resolve data-deficient problems and aims to generalize unfamiliar tasks considering prior knowledge of trained agents and few test samples in a rapid manner [30]. At present, meta-learning algorithms are usually applied in the field of few-shot learning. The purpose of the trained model in meta-learning is to quickly learn a new task from small quantities of fresh data, and the model learns from a variety of tasks trained by a meta-learner [31,32].

There are three types of meta-learning approaches: model-based, metric-based, and optimization-based approaches. The model-based approaches integrate the current dataset $\mathcal{D}$ into an activation state, and predictions for test data are derived based on this state [33]. The goal of metric-based approaches is to learn the metric or distance function between different samples [17,34,35]. Optimization-based approaches resolve the inner-level task as an optimization problem and extract meta-knowledge w to increase the optimization performance [11,30,36,37].

Further advances were made by using gradient-based meta-learning algorithms that expressly optimize themselves with a certain amount of data points for quick adaption. One popular approach is to learn a parameter initialization and optimizer parameterized as neural networks for quick adaptation, notably the Meta-LSTM [30] approach. The model agnostic meta-learning (MAML) algorithm [11] improves Meta-LSTM performance, as the parameters of the MAML are specifically developed to generate high overall performance with a small number of gradient steps and a few training samples from a new task. Meta-SGD [38] is an improved version of MAML, and Meta-SGD uses the SGD [39] optimizer to optimize the internal learning rate of MAML, thereby improving the performance of the model.

Few-shot meta-learning is designed to train a model that can be quickly adapted with a few training samples. The meta-learning model is trained on a series of tasks in the meta-learning training stage in order to achieve rapid adaptation to new tasks with a limited sample size. Thus, we believe the rapid generalization of remote sensing image scene classification in different cities can be viewed as a meta-learning problem. We expect our model to quickly generalize to other unknown urban areas using data of known cities. Different cities are defined as different tasks, and only a small number of remote sensing scene images of each city are used for model training.

We define a base model as f that represents a backbone network to maps input x to output a, where x represents the input images, and a represents extracted features. In the meta-training phase, we define a single task as $\mathcal{T}$, which is sampled in $p(\mathcal{T})$. In our meta-learning scenario, we expect our model to quickly adapt to the distribution on the task $p(\mathcal{T})$. Remote sensing data in different $\mathcal{T}$ are obtained from different cities. Figure 1 presents the schematic diagram of the data division from a 5-way 1-shot remote sensing scene classification task.

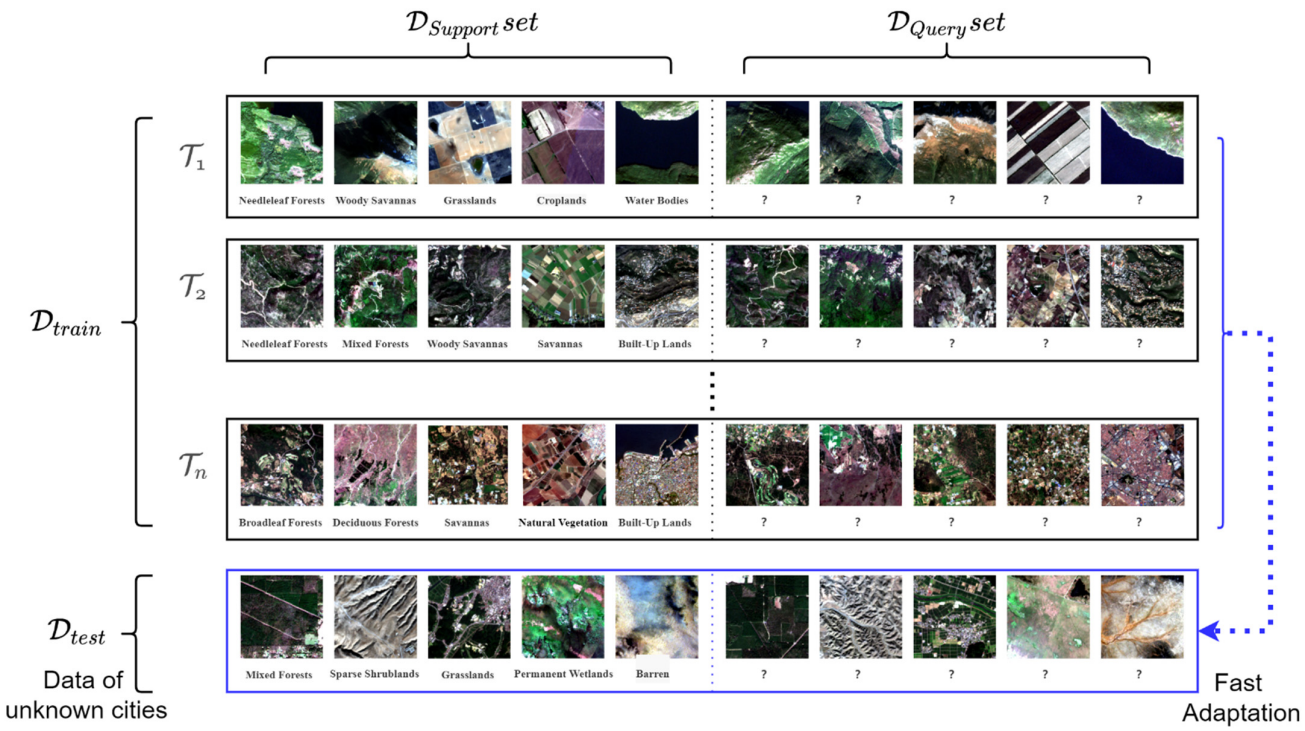


Figure 1. The schematic diagram of the data division from a 5-way 1-shot few-shot remote sensing scene classification. In $\mathcal{D}_{train}$, each task $\mathcal{T}$ represents a dataset extracted from a certain city. We use the $\mathcal{D}_{Support}$ set to learn the internal optimization weights, and the $\mathcal{D}_{Query}$ set to update the weights. Further, we use the self-supervised embedding optimization module to balance the requirements for feature classification tasks between multiple images collected at different times and places and use the $\mathcal{D}_{test}$ to test the generalization performance of the model to unknown cities.

The training of few-shot learning involves a dataset that is split into two parts, a $\mathcal{D}_{Support}$ set for learning and a $\mathcal{D}_{Query}$ set for training or testing, i.e., $\mathcal{D} = \{\mathcal{D}_{Support}, \mathcal{D}_{Query}\}$. A K-shot and N-class classification task suggests that a total of N classes are present in one task, with each class containing a total of K samples. A task $\mathcal{T}$ consists of a support dataset $\mathcal{D}_{Support}$ to adapt the model parameters to the particular task and a query dataset $\mathcal{D}_{Query}$ to evaluate the performance. Following the meta-training phase, a new task starts to sample from $p(\mathcal{T})$ as the meta-validation dataset. Meta-learning internal optimization is applied to quickly adjust the parameters of the model to adapt to unknown urban scenes in the meta-validation dataset.

2.4. The Proposed Method

The optimization-based meta-learning algorithms, aiming to obtain an initialization model or gradient descent direction through the meta-learning processing, are an important branch in the field of meta-learning. Example algorithms include MAML and Meta-LSTM. However, these methods cannot balance the requirements for feature classification tasks between multiple images collected at different times and places, leading to a decline in generalization ability. The key explanation is that, regardless of how general the initial model is expected to be, the model is still trained on a limited set of samples. Our model, however, should own adaptation and generalization capability to execute new tasks with unseen urban areas. For this consideration, the main contribution of this paper is to design a self-supervised embedding optimization (SEO) based on optimization-based meta-learning by designing a loss function between multiple tasks. The proposed meta-learning fast adaptation with self-supervised embedding optimization (Meta-FSEO) model is expected to effectively balance the requirements for feature classification

between multiple tasks in the $\mathcal{D}_{Query}$ set optimization stage, leading to improved generalization capability. Figure 2 and Algorithm 1 present the specific process of our proposed Meta-FSEO.

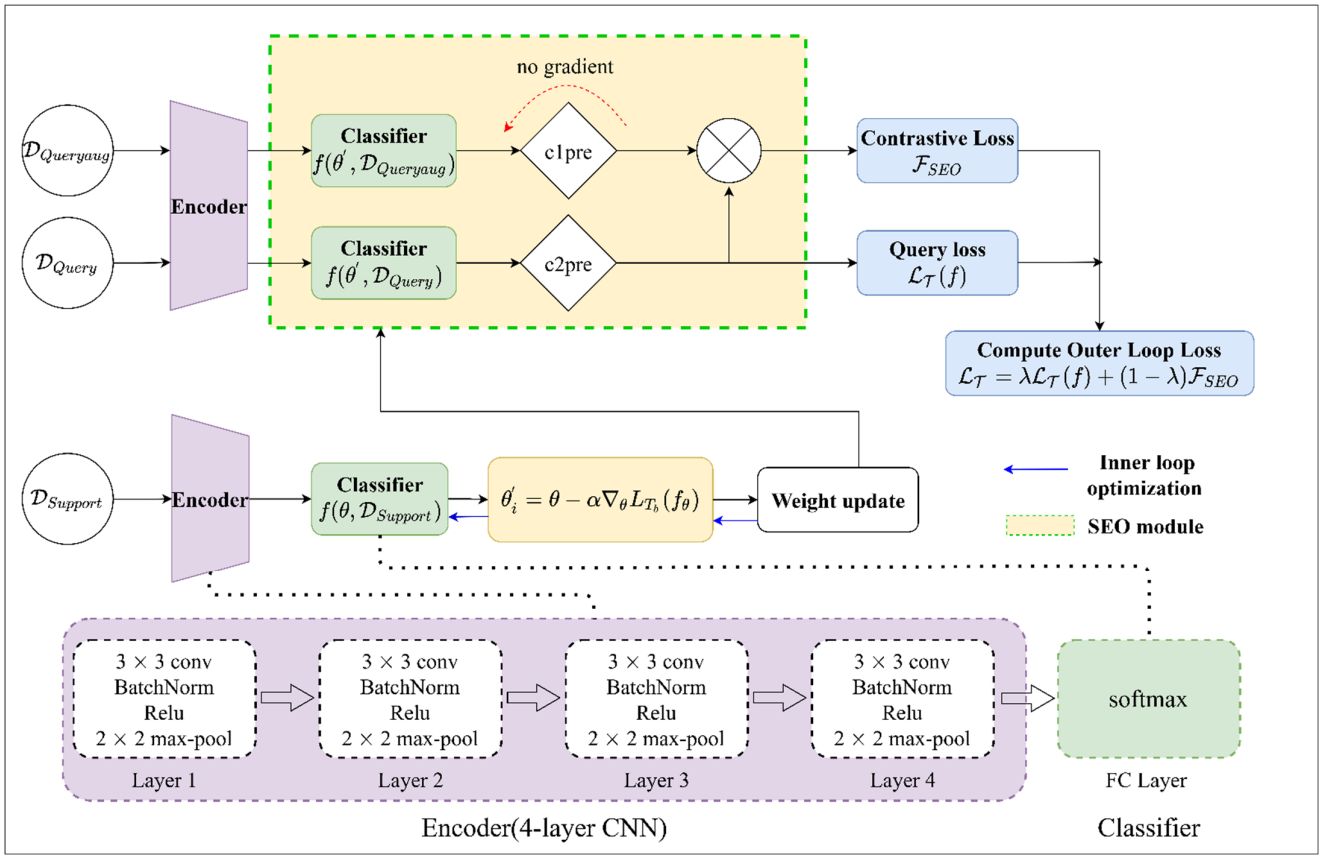


Figure 2. The proposed Meta-FSEO. Starting from the lower-left corner, the divided support dataset first passes through a backbone encoder and classifier (encoder represents a 4-layer CNN and classifier represents a classification output result) and updates θ to θ_i through an internal optimization of meta-learning. We pass the $\mathcal{D}_{Query}$ set and $\mathcal{D}_{Queryaug}$ set (after data enhancement) into our self-supervised embedding optimization module for the next update (diamond represents the prediction result, a cross in a circle represents Equation (3)), combine contrast loss and the cross-entropy loss to obtain the final loss function, and perform external optimization. The encoder is composed of a four-layer convolutional network structure that contains a 3 × 3 convolution with 48 filters, followed by batch normalization with momentum set to 0.1, a ReLU nonlinearity, a 2 × 2 max-pooling, a linear layer, and a SoftMax layer. The padding and stride are both set to 1, except for the 2 × 2 max-pooling, whose stride and padding are set to 2 and 0, respectively. In batch normalization, the momentum is set to 0.1. The epsilon, a value in the denominator for numerical stability, is set to 0.0001.

Our data are processed and divided into $\mathcal{D}_{Support}$, $\mathcal{D}_{Query}$, and $\mathcal{D}_{Queryaug}$, following a meta-learning standard. The $\mathcal{D}_{Queryaug}$ dataset is derived from $\mathcal{D}_{Query}$ after data enhancement procedures that include random horizontal flip and a random vertical flip. An encoder (4-layer CNN) is employed to encode the $\mathcal{D}_{Support}$, $\mathcal{D}_{Query}$, and $\mathcal{D}_{Queryaug}$, as shown in Figure 2. Entering the internal optimization stage, a SoftMax function classifies the features extracted from the $\mathcal{D}_{Support}$ dataset and adopts a cross-entropy loss $\mathcal{L}_{\mathcal{T}}(f)$ to optimize network parameters (Algorithm 1):

$$\mathcal{L}_{\mathcal{T}}(f) = \sum_{x,y}^{\mathcal{T}} y\log\, f(x) + (1-y)\log\,(1-f(x)) \tag{1}$$

where x represents the input image, f represents the prediction function, and y denotes the ground-truthing value.

The proposed Meta-FSEO model uses a gradient descent-based inner optimization to update the $\mathcal{D}_{Support}$ dataset with new parameters θ':

$$\theta_i' = \theta - \alpha \nabla_\theta L_{T_b}(f_\theta) \tag{2}$$

where α is the inner meta step size.

For some parameterized models $f_{\theta'}$ in the $\mathcal{D}_{Support}$ set, we use the parameters of this model $f_{\theta'}$ to calculate $\mathcal{F}_{SEO}$, $\mathcal{D}_{Query}$, and $\mathcal{D}_{Queryaug}$ encoded by p and n, respectively. We calculate the cosine similarity for the upper and lower batches and obtain the $N \times N$ matrix. The diagonal position of each row represents the similarity between p and n, while the rest positions represent the similarity between p and $N-1$ negative examples. We use SoftMax classification for each row with cross-entropy loss to achieve contrastive learning:

$$\mathcal{F}_{SEO} = -\log \frac{\exp\left(p \cdot n/th\right)}{\sum_{i=0}^{N} \exp\left(p \cdot n_i/th\right)} \tag{3}$$

where th is a temperature hyperparameter.

For the external optimization, we hope to balance the preferences among tasks. The final external optimization loss is the weighted sum of the cross-entropy loss [40] of multiple tasks and the contrast loss [39,40] of $\mathcal{D}_{Query}$ and $\mathcal{D}_{Queryaug}$:

$$\mathcal{L}_{\mathcal{T}} = \lambda \mathcal{L}_{\mathcal{T}}(f) + (1-\lambda)\mathcal{F}_{SEO} \tag{4}$$

Finally, a gradient descent update is performed between multiple tasks:

$$\theta \leftarrow \theta - \gamma \nabla_\theta \sum_{T_b}^{p(T)} \mathcal{L}_{\mathcal{T}} \tag{5}$$

where γ is the meta step size.

Algorithm 1: Meta-FSEO Algorithm.

Input:

Base model function f and initialisation parameters θ, Self-supervised Embedding Optimization fuction $\mathcal{F}_{SEO}$ and parameters w, step size hyperparameters α, β, γ

Randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_b \sim p(\mathcal{T})$
  for all $\mathcal{T}_b$ **do**
    for i in rang(N) **do**
      Sample K datapoints $\mathcal{D}_{Support} = \{x_S^b, y_S^b\}$ from $\mathcal{T}_b$
      Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_b}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_b}$ in Equation (1)
      Inner loop optimization in support sets: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_b}(f_\theta)$
    end for
    Sample K datapoints $\mathcal{D}_{Query} = \{x_Q^b, y_Q^b\}$ and $\mathcal{D}_{Queryaug} = \{x_{Qaug}^b, y_{Qaug}^b\}$ from $\mathcal{T}_b$
    Compute $\mathcal{F}_{SEO}$ with transformer net
    $\theta_{N+1}' = \theta_N - \beta \nabla_{\theta_N} \mathcal{F}_{SEO}(f_{\theta_N})$
    Update $\theta \leftarrow \theta - \gamma \nabla_\theta \sum_{\mathcal{T}_b}^{p(\mathcal{T})} \mathcal{L}_{\mathcal{T}}$ using $\mathcal{L}_{\mathcal{T}}$ and $\mathcal{D}_{Query}$ in Equation (4)
  end for
end while

More methodological details regarding the proposed Meta-FSEO can be found in Algorithm 1. In the next section, we describe the experimental results of Meta-FSEO in detail.

3. Results

We evaluate the proposed Meta-FSEO model on Sentinel-1/2 multi-spectral (SEN12MS), a large remote sensing scene classification dataset. We process the dataset and compare our proposed Meta-FSEO with two meta-learning algorithms (i.e., MAML, Meta-SGD). In this section, we introduce SEN12MS and preprocessing steps. We detail the structure of a network and the settings for hyperparameters and evaluate the accuracy of classification results from the proposed method and the other tested algorithms against the ground-truth data.

3.1. Datasets and Preprocessing

The SEN12MS [41] is a new classification and segmentation dataset of satellite images with global distribution. SEN12MS contains 180,662 image patches triplets with dual-pol synthetic aperture radar (SAR), multi-spectral Sentinel-2, and MODIS land cover maps. The sample distance to each patch of all climate seasons is 10 m, and the size of each patch is 256 × 256 px. Images in SEN12MS are distributed in 125 cities around the world with 17 different scene categories, i.e., Evergreen Needleleaf Forests, Evergreen Broadleaf Forests, Deciduous Needleleaf Forests, Deciduous Broadleaf Forests, Mixed Forests, Closed (Dense) Shrublands, Open (Sparse) Shrublands, Woody Savannas, Woody Savannas, Grasslands, Permanent Wetlands, Croplands, Cropland/Natural Vegetation Mosaics, Permanent Snow and Ice, Barren, and Water Bodies.

In the original dataset, each image has an overlap of 50%. We crop the overlapping section of each image. After cropping, the size of each image was 128 × 128 px without changing the number of images. Following the data division method in [18], we divide the images from 125 globally distributed cities in SEN12MS into a meta-train set, meta-val set, and meta-test set according to the ratio of 3:1:1 (Figure 3). The model training and validation are performed on the meta-train set and meta-val set, while the final model is evaluated on the meta-test. In the evaluation, we use images in cities that are not included during the training process so that the model can learn how to generalize towards unknown urban areas.

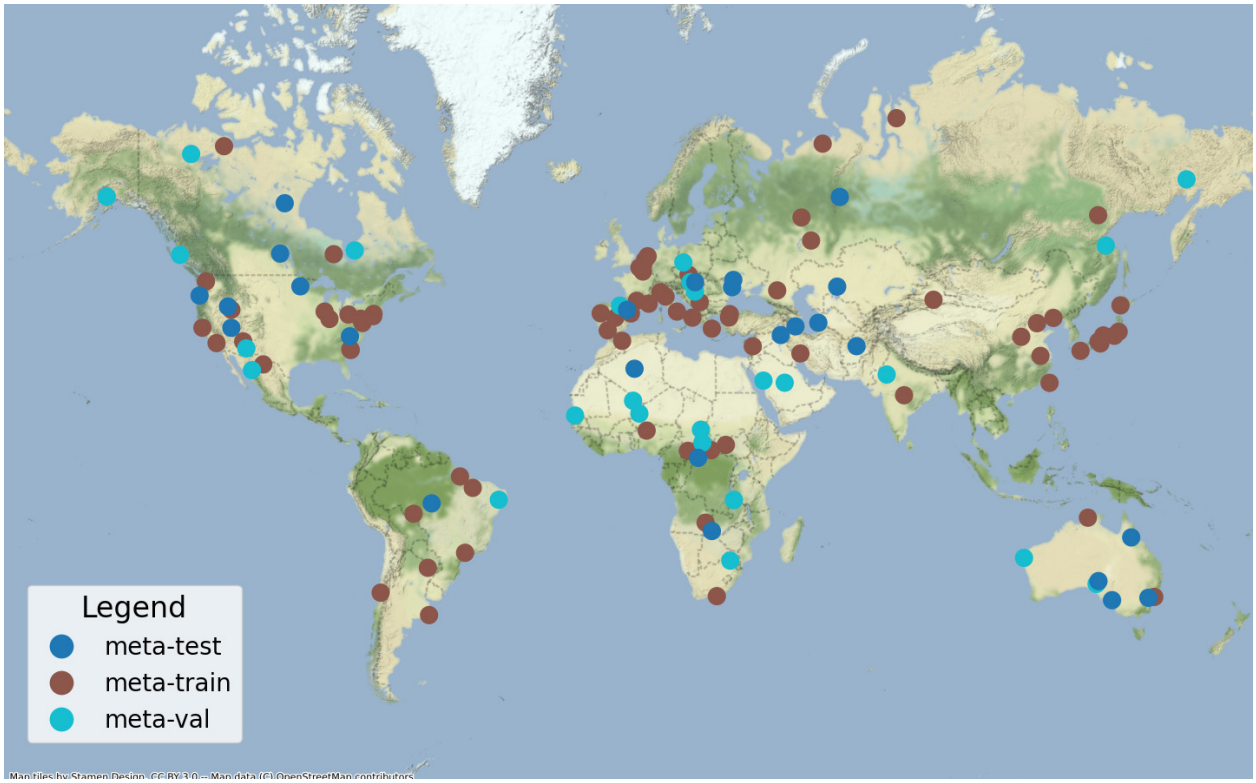


Figure 3. The SEN12MS dataset is a publicly available remote sensing dataset that includes 125 city regions all around the world. We divided the 125 urban areas in SEN12MS into meta-train, meta-val, and meta-test at a ratio of 3:1:1.

During model training, data from each city are defined as a task sampling pool, and data generation tasks are randomly collected in different cities. In our experiment, all tasks sample from five categories, and each category samples 1–5 shots (one shot represents a 128 × 128 image) as five experiments. Figure 2 shows an example of the five-way, one-shot scenario. Note that each task $\mathcal{T}$ selects five categories arbitrarily from the data of a certain city, and each category randomly samples two images, which is the sum of the number of each category selected in $\mathcal{D}_{Support}$ set and the number of each category in the $\mathcal{D}_{Query}$ set. When the number of scene categories is less than five in the original data (not enough to sample a meta-learning task), we exclude data from these cities. We observe that scenes of the same category greatly differ among cities, especially whenthere is a long geographic distance between cities. We expect our model to quickly generalize classification tasks to unknown cities by taking advantage of only a few samples.

3.2. Hyperparameters Details

In this section, we present network structure, experiments, and hyperparameter settings in detail. For a fair comparison, we use a four-layer convolutional network structure to extract features for all comparative models. The four-layer convolutional network structure contains a 3 × 3 convolution with 48 filters, followed by batch normalization, a ReLU nonlinearity layer, 2×2 max-pooling, a linear layer, and a softmax layer. The padding and stride are both set to 1, except for the 2 × 2 max-pooling, whose stride and padding are set to 2 and 0, respectively. In batch normalization, the momentum is set to 0.1. The epsilon, a value in the denominator for numerical stability, is set to 0.0001. All comparative models use a cross-entropy loss to measure the difference between the predicted value and the ground truth. Each gradient in N-way, K-shot classification is computed using a batch size of N × K examples (N = 5 and K = (1,2,3,4,5)). As shown in Table 1, the number of iterations for internal optimization is set to 5, and the inner meta step size α is

set to 0.01. We set the meta step size β to 0.01 and the learning rate γ to 0.001 for optimizing the $\mathcal{D}_{Query}$ dataset. For the loss function $\mathcal{L}_{\mathcal{T}}$, the balance parameter λ is initially set to 0.7 (detailed discussion of the setting of λ can be found in Section 4.3). For all models, we use the Adam optimization with a weight decay of 0.0001. All models are trained for 75,000 iterations with a total of 150 epochs on four NVIDIA Pascal Titan XP GPU (12G $\times$ 4).

Table 1. Hyperparameter settings in our model.

Name	Parameter
inner meta size a	0.01
meta step size	0.01
learning rate	0.001
weight decay	0.0001
balance parameter	0.7
iterations	75,000
epoch	150

3.3. Classification Accuracy

We compare our proposed method, i.e., Meta-FSEO, with a transfer learning method and two optimization-based meta-learning methods, i.e., MAML and Meta-SGD. For a fair comparison, all models are with the same backbone, i.e., a four-layer CNN network structure described in Section 3.2. For the transfer learning method, we pre-train it on a large dataset (ImageNet 2010), fine-tune it on the meta-train set, select the model with the highest verification accuracy in the meta-val set, and evaluate its performance on the meta-test set. For the meta-learning method, we select MAML and Meta-SGD algorithms as the comparative algorithm. MAML is a classic and popular algorithm in the field of meta-learning. Meta-SGD is an improved version of MAML with optimized gradient descent in the learning rate of the inner loop. Our proposed Meta-FSEO model is also an optimization-based meta-learning method directly trained on the meta-train dataset. We further evaluate its performance on the meta-test set. Note that Meta-FSEO does not require a pre-trained model.

We evaluate the performance of these models using classification accuracy, which is defined as:

$$Accuracy = \frac{1}{\mathcal{T}} \sum_{i=1}^{\mathcal{T}} \frac{r^i}{S} \tag{6}$$

where $\mathcal{T}$ is the number of tasks, r^i is the number of samples correctly predicted for the i-th task, and S is the total number of samples for this task.

All experiments in this study correspond to five categories. When evaluating the performance of the model, we expect our model to obtain a satisfactory performance on samples from cities not included in the training phase. To evaluate the generalization ability of the model, we randomly select a small number of test data from the meta-test dataset. To reduce the model's preference towards certain tasks, we set up 600 experiments in the testing phase by randomly sampling tasks in the meta-test dataset. The final test accuracy is the average of accuracy values from all tasks.

The comparison between the proposed Meta-FSEO and two meta-learning-based methods (MAML and Meta-SGD) is shown in Figure 4. We observe that the performance of our method boosts in a more significant manner with the increase in labeled samples compared with the other two algorithms. From one-shot to two-shot, the accuracy of our model increases more rapidly, suggesting that our model is more adaptable to a smaller sample size. In addition, we present the confusion matrix of Meta-FSEO. Figure 5 shows

the confusion matrix classification results of the one-shot to five-shot in Meta-FSEO, where the entries in row i and column j represent the probability that the test image of category i is classified as category j. From the confusion matrix of one-shot to five-shot, it can be seen that the prediction accuracy of the proposed Meta-FSEO model is gradually improved, suggesting that the number of samples has a great influence on the accuracy of the model. Experiments with fewer samples (e.g., one-shot and two-shot) have larger accuracy differences among different categories. In comparison, experiments with more samples (e.g., five-shot) present balanced accuracy among different categories.

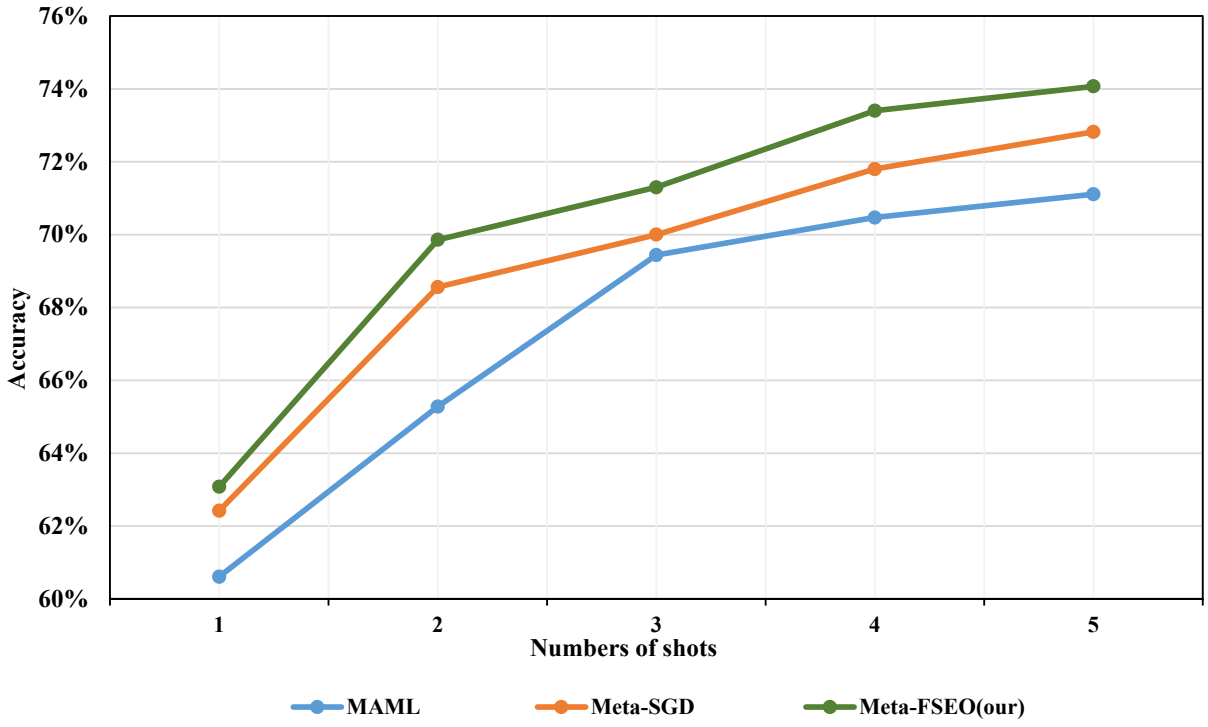
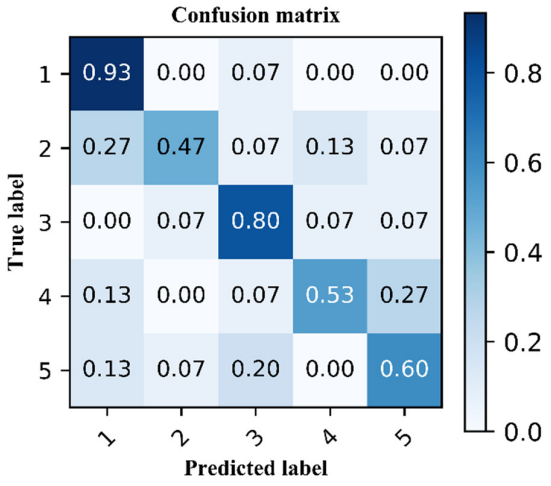
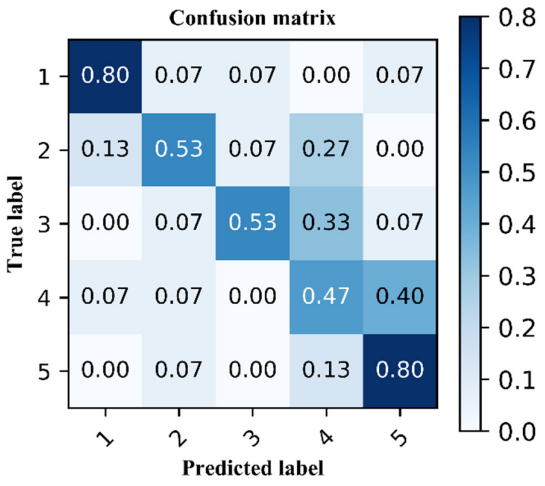


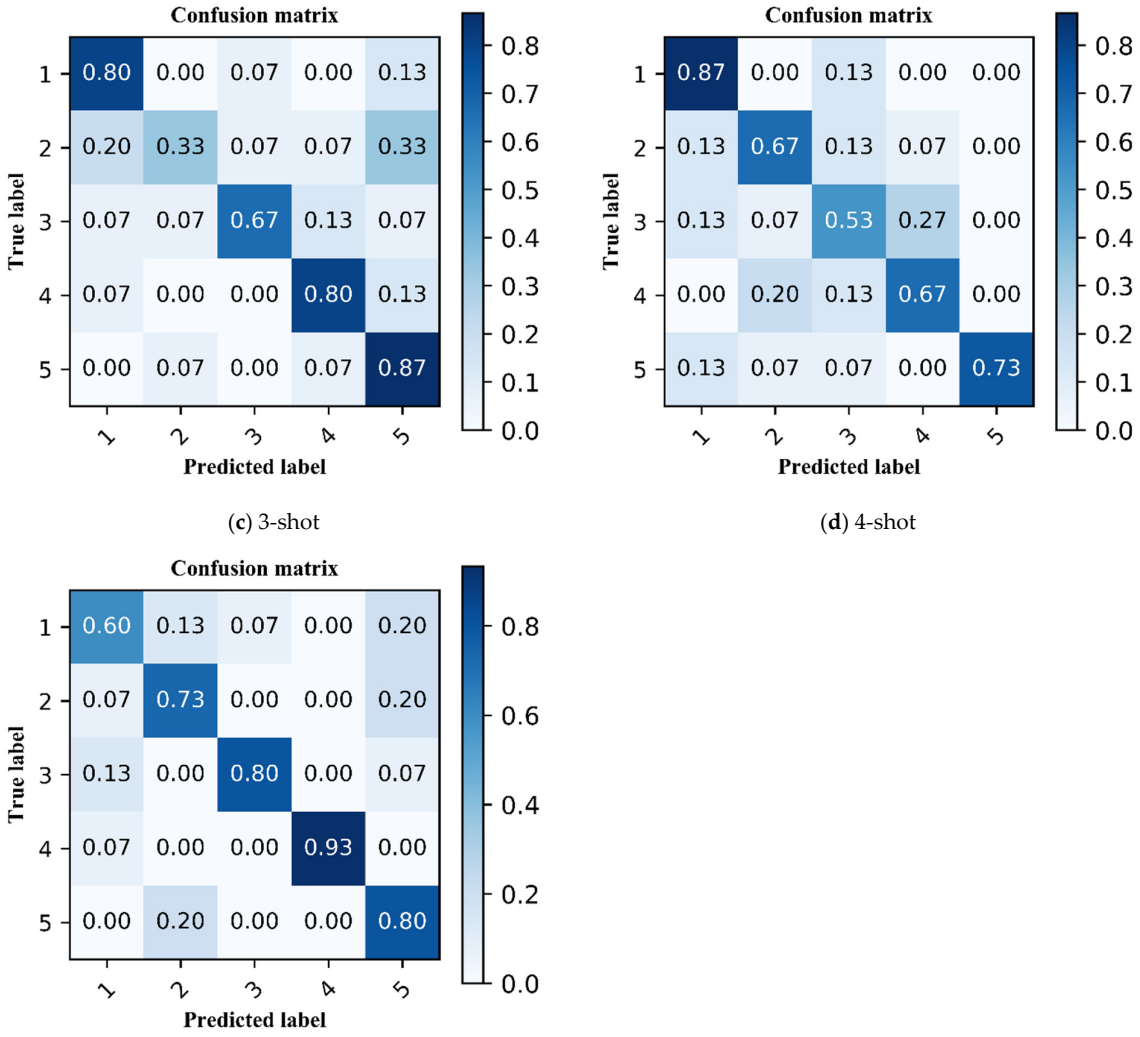
Figure 4. Classification accuracy of the proposed Meta-FSEO compared with MAML and Meta-SGD. The *x*-axis suggests the number of shots of the meta-test. Given 1–5 labeled samples, our proposed Meta-FSEO outperforms the MAML and Meta-SGD.



(**a**) 1-shot



(**b**) 2-shot



(**c**) 3-shot (**d**) 4-shot

(**e**) 5-shot

Figure 5. Confusion matrices of the proposed Meta-FSEO from 1-shot to 5-shot, where the entries in row i and column j represent the probability that the test image of category i is classified as category j.

4. Discussion

In this study, we test the performance of the trained model on new remote sensing scenes from different cities, which are not included in the training set and expect the trained model to achieve great classification performance with just a few labeled samples in remote sensing scenes from unknown cities. In this section, we analyze the effectiveness of the transformers structure, conduct an ablation experiment to investigate the effectiveness of the self-supervised embedding optimization (SEO) module, and present a loss analysis.

4.1. The Effect of the Transformers in Network Architecture

This section focuses on investigating the usability of the transformers in the backbone network. We aim to incorporate transformers into the backbone network to improve the performance of the model. We use a multi-head self-attention mechanism to replace the space convolution layer [24]. The use of self-attention in the backbone is flexible in terms of input resolution requirements (the input resolution of our model is 128 × 128). Considering that self-attention is computationally intensive demanding during its execution,

we add self-attention to low-resolution feature maps of the backbone. We add multi-head self-attention to the last layer of the backbone to form a CNN + transformers structure. In order to make the attention operation position aware, we adopt the 2D relative position self-attention implementation from [28]. Table 2 shows the results of different models under five-way one-shot and five-way five-shot scenarios. We use a standard four-layer convolutional network (described in Section 3.2) and a standard four-layer convolutional network with the transformers as the backbone in all comparisons. For the fine-tuning model, we train the model on ImageNet-2010, with the learning rate initialized to 0.01 (reducing three times before training stops). We train the neural network for 90 epochs on the training dataset of 1.2 million images and fine-tune the resulted model on the SEN12MS dataset. For MAML, we follow the training strategies from [11] with the learning rate α set to 0.01 and the number of gradient steps set to five. For Meta-SGD, we follow the training strategies from [38] with the learning rate α set to 0.01 and the number of gradient steps set to one. The results suggest that the proposed Meta-FSEO achieves the highest accuracy in both scenarios, while the performance of the fine-tuning model based on the transfer learning method is relatively poor. Meta-SGD, with a standard four-layer CNN structure, achieves an accuracy of 62.42% and 72.82% under five-way one-shot and five-way five-shot scenarios, respectively, surpassing popular meta-learning algorithms, i.e., MAML and Meta-SGD. The proposed Meta-FSEO, incorporated with a transformer, obtains an improved performance, achieving an accuracy of 63.08% and 74.29% under five-way one-shot and five-way five-shot scenarios, respectively. We further notice that the application of the transformers module leads to a more notable improvement in our proposed Meta-FSEO compared to MAML and Meta-SGD.

Table 2. The classification accuracy of different models on the SEN12MS dataset. The accuracy of Meta-FESO achieves the highest accuracy in one-shot and five-shot scenarios, surpassing the fine-tuning model, and two meta-learning models, i.e., MAML and Meta-SGD. The inclusion of the transformers module further improves performance for all models.

Model	Backbone	5-Way Accuracy	
		1-Shot	5-Shot
Transfer Learning (fine-tuning)	4-layer CNN	35.20 ± 0.603	53.90 ± 0.720
	4-layer CNN + Transformers	35.86 ± 0.554	54.79 ± 0.690
MAML	4-layer CNN	60.02 ± 0.495	69.44 ± 0.460
	4-layer CNN + Transformers	60.61 ± 0.488	70.00 ± 0.488
Meta-SGD	4-layer CNN	61.22 ± 0.486	71.11 ± 0.486
	4-layer CNN + Transformers	61.72 ± 0.486	72.48 ± 0.446
Meta-FSEO (ours)	4-layer CNN	62.42 ± 0.484	72.82 ± 0.445
	4-layer CNN + Transformers	63.08 ± 0.480	74.29 ± 0.437

4.2. Ablation Analysis

In this section, we conduct an ablation experiment to investigate the effectiveness of the self-supervised embedding optimization (SEO) module. Figure 6 shows the results of the ablation study of SEO. We notice that when the SEO module is ablated, the performance of the model drops significantly in all shots. As SEO is removed, the model tends to prioritize specific tasks, leading to reduced adaptation and generalization capability. We also notice that the SEO module plays an increasingly important role in model performance with the growth of shots. For instance, under the one-shot scenario, the accuracy of the model drops by 2% when SEO is removed. Under the five-shot scenario, however, the accuracy of the model drops by 5% when SEO is removed.

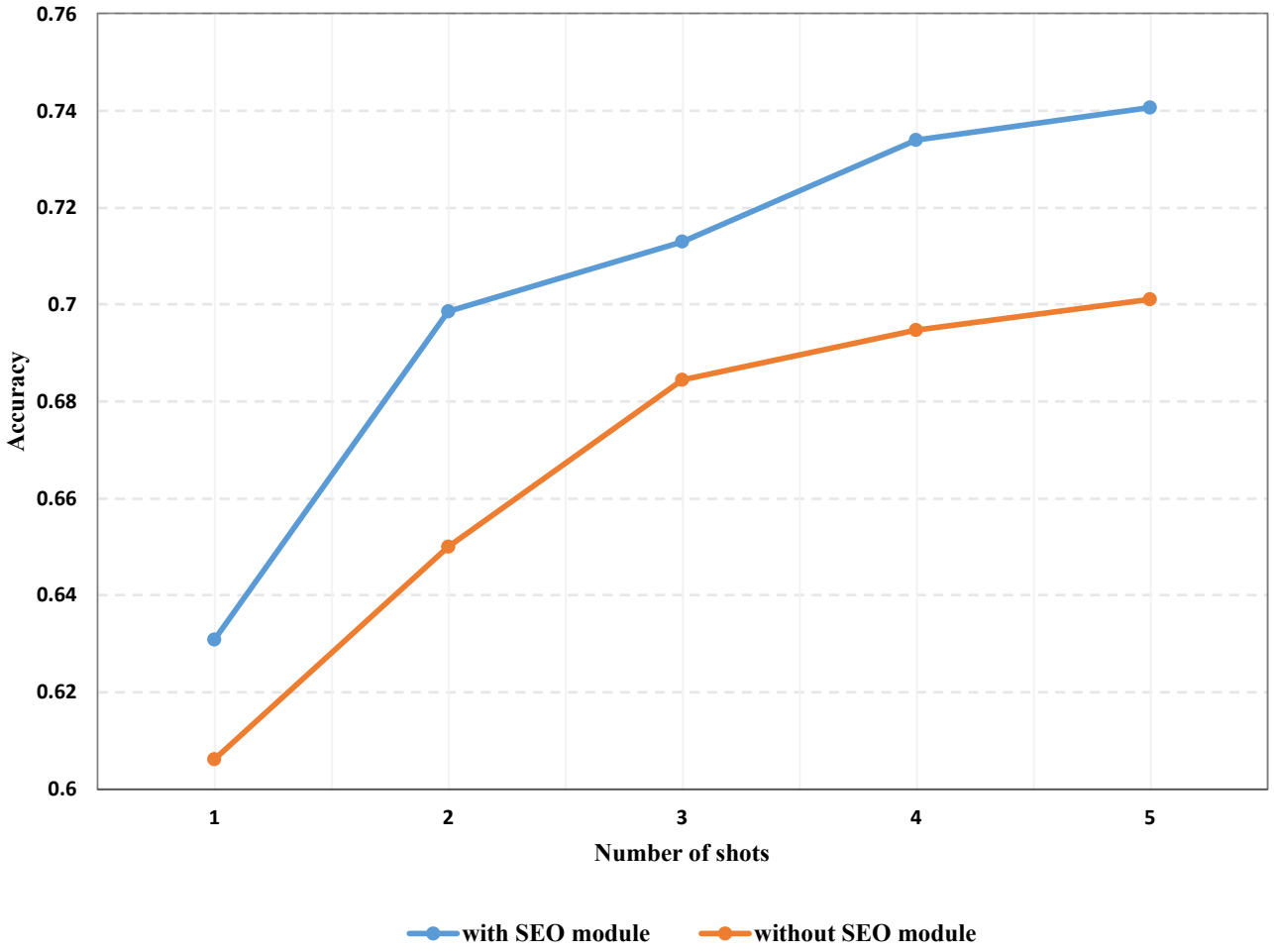


Figure 6. Results of the ablation study of the SEO module. After the SEO module is ablated, the performance of the model drops significantly in all shots.

4.3. Loss Analysis

In this section, we analyze the impact of hyperparameter λ (see Equation (4)) in our designed loss. Hyperparameter λ serves as a weighting factor that balances the cross-entropy loss of multiple tasks and the contrast loss of $\mathcal{D}_{Query}$ and $\mathcal{D}_{Queryaug}$. We investigate the value settings of λ in a range of (0, 1) with 0.1 as an interval. Figures 7 and 8 present the accuracy and standard deviation of our model under the one-shot and five-shot scenarios when λ takes different values. We noticed that when λ reaches 0.7, the classification accuracy of one-shot reaches the highest with the smallest standard deviation, suggesting a strong generalization ability of the model. With a gradual increase in λ starting from 0.1 (leading to an increased weight of the contrast loss), the accuracy of the model gradually increases with a decreased standard deviation, suggesting that the contrast loss has a positive effect on the generalization ability of our model. When λ grows to 0.7, the model reaches its maximum performance (under the one-shot scenario). However, as λ continues to grow, the model performance shows a declining tendency, indicating the balancing role of the contrast loss, as an excessively large λ tends to overbalance certain tasks, negatively impacting the generalization ability of our model. When λ takes different values, the five-shot scenario and the one-shot scenario of our model have similar trends in the distribution of accuracy and standard deviation. We notice that when λ reaches

0.8, the classification accuracy reaches the highest with the smallest standard deviation (under the five-shot scenario), suggesting a strong generalization ability of the model.

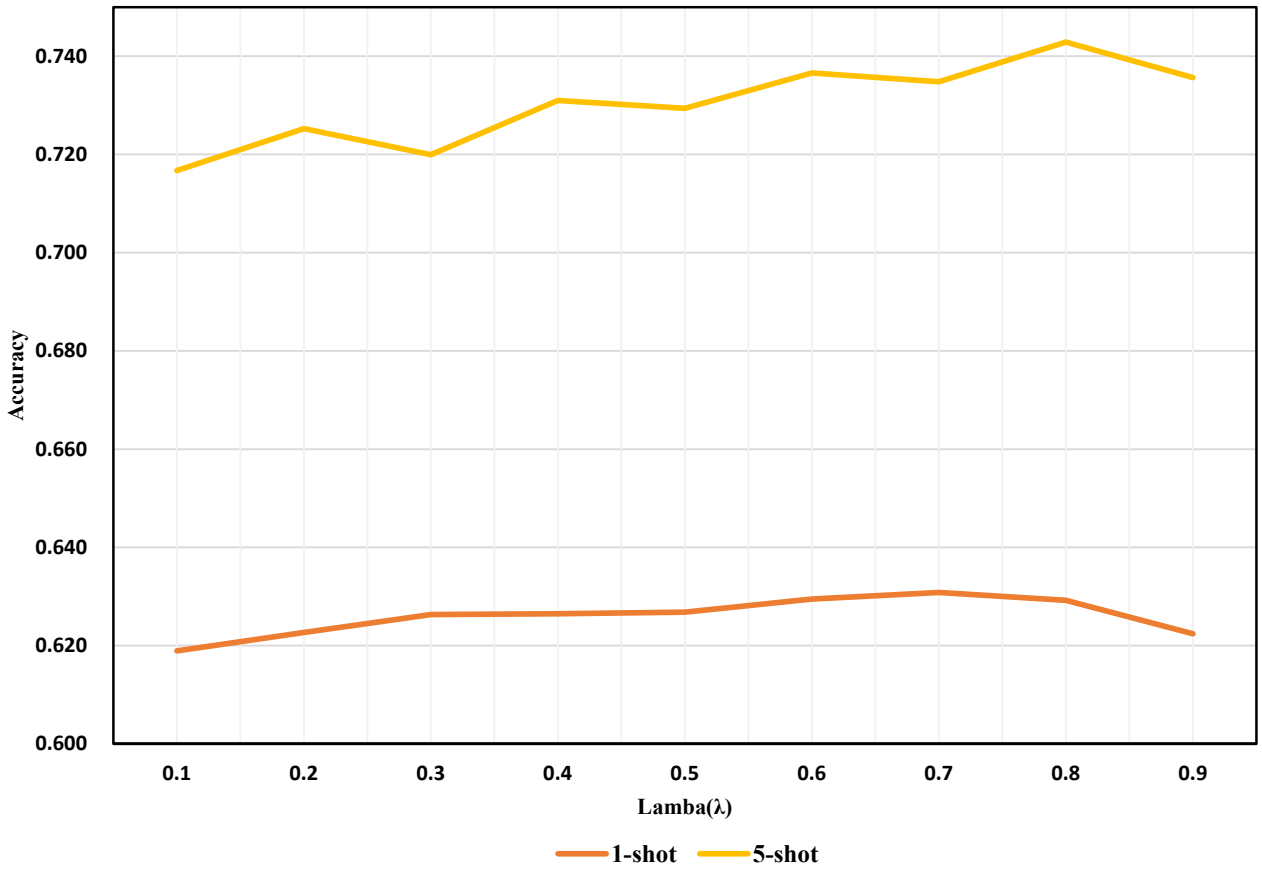


Figure 7. Model's accuracy with different settings of hyperparametric λ in 1-shot and 5-shot scenarios. The x-axis shows the different values of λ, and the y-axis shows the model's accuracy. The orange curve represents the classification accuracy of the 1-shot scenario. The model's accuracy improves as the λ increases, reaching a maximum with λ of 0.7. However, the model's accuracy starts to decrease with an excessively large λ. The yellow curve represents the classification accuracy of the 5-shot scenario. Similarly, the model's accuracy improves as the λ increases, reaching a maximum with λ of 0.8. However, the model's accuracy starts to decrease with an excessively large λ.

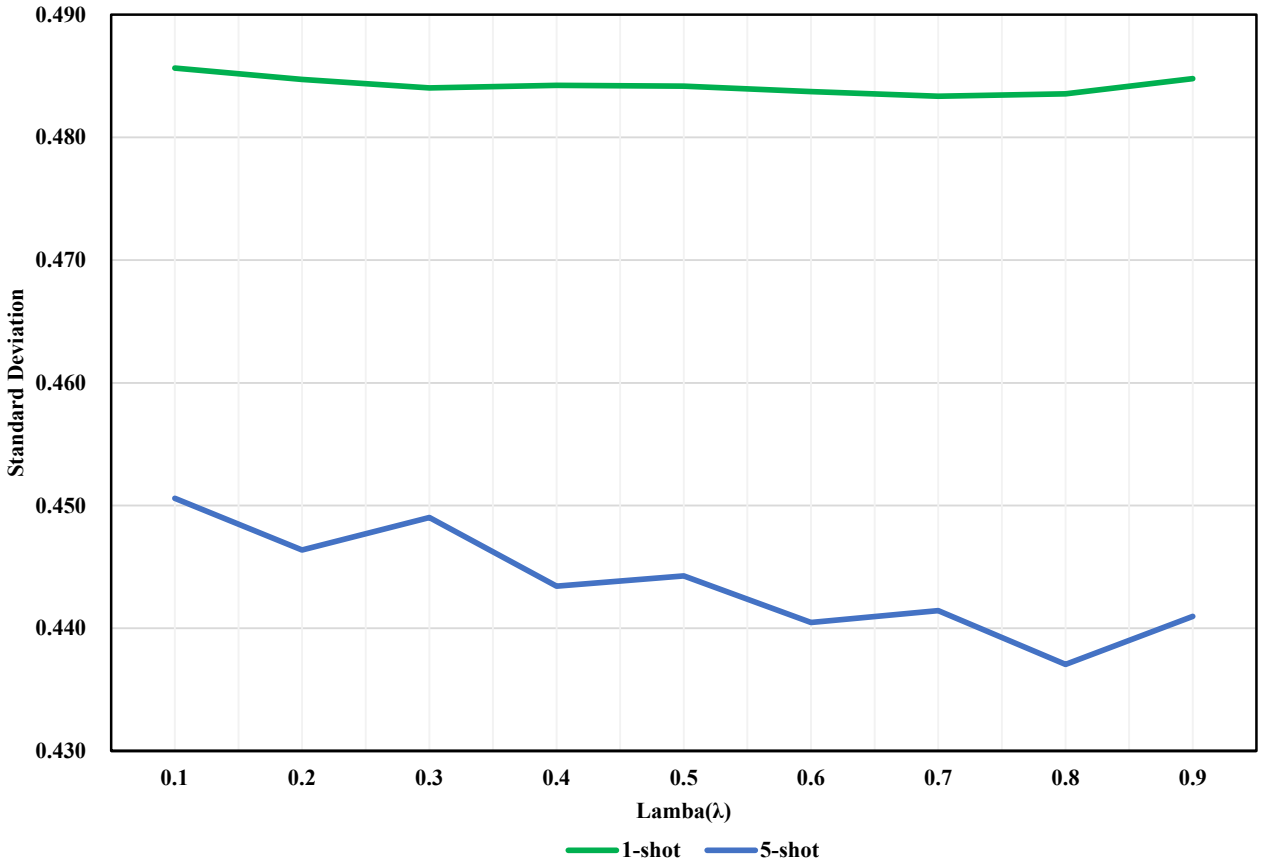


Figure 8. The standard deviation of model's performance with different settings of hyperparametric λ in 1-shot and 5-shot scenarios. The x-axis shows the different values of λ, and the y-axis shows the model's standard deviation. The green curve represents the standard deviation of the 1-shot scenario. The model's standard deviation decreases as λ increases, reaching a minimum with λ of 0.7. However, the standard deviation of the model's accuracy starts to increase with an excessively large λ. The blue curve represents the standard deviation of the 5-shot scenario. Similarly, the model's standard deviation decreases as λ increases, reaching a minimum with λ of 0.8. However, the standard deviation of the model's accuracy starts to increase with an excessively large λ.

5. Conclusions

In this study, we propose a meta-learning algorithm called Meta-FSEO to improve the generalization performance of remote sensing scene classification in multiple urban conditions under few-shot scenarios. The proposed Meta-FSEO allows quick generalization to the data from unknown cities by training on the data from known cities according to task-level samples. It not only realizes adaptive optimization of the support set but also optimizes the query set through a self-supervised embedding optimization (SEO) module. In addition, we design a new loss function that combines the contrast loss and the cross-entropy loss.

To verify the performance of our model, we test the proposed Meta-FSEO as well as other comparative methods on SEN12MS, a popular remote sensing classification and segmentation dataset with globally distributed satellite images. The results show that our model has great superiority under all scenarios, compared to a fine-tuning method and two meta-learning-based methods, i.e., MAML and Meta-SGD. We also perform an ablation investigation on the SEO module. The results show that, under one-shot and five-shot scenarios, the performance of the model drops significantly in all shots when the proposed SEO module is ablated, suggesting the important role of SEO in the model's adaptation

and generalization capability. Furthermore, we investigate the effect of hyperparameter λ of our designed loss in model performance and derive an optimized λ. At present, we only verify our model in remote sensing scene recognition tasks. In the future, we plan to apply the proposed Meta-FSEO to other urban land surface segmentation tasks.

Author Contributions: Conceptualization, Y.L. and Z.S.; methodology, Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L.; investigation, Y.L. and B.C.; resources, Y.L. and S.P.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and X.H.; visualization, Y.L.; supervision, Z.S.; project administration, Z.S.; funding acquisition, Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China with grant number 2018YFB2100501, the Key Research and Development Program of Yunnan province in China with grant number 2018IB023, the National Natural Science Foundation of China with grant numbers 42090012, 41771452, 41771454, and 41890820, consulting the research project of Chinese Academy of Engineering with grant number 2020ZD16.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available in a publicly accessible repository. The data presented in this study are openly available in https://mediatum.ub.tum.de/1474000 (2021/4/5), reference number [41].

Acknowledgments: The authors are sincerely grateful to Steve McClure for revised the grammatical errors in the paper, and the editors, as well as the anonymous reviewers, for their valuable suggestions and comments that helped us improve this paper significantly.

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

References

1. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36, doi:10.1109/MGRS.2017.2762307.
2. Zhang, L.; Zhang, L.; Du, B. Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40, doi:10.1109/MGRS.2016.2540798.
3. Shao, Z.; Cai, J.; Fu, P.; Hu, L.; Liu, T. Deep Learning-Based Fusion of Landsat-8 and Sentinel-2 Images for a Harmonized Surface Reflectance Product. *Remote Sens. Environ.* **2019**, *235*, 111425, doi:10.1016/j.rse.2019.111425.
4. 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.* **2018**, *56*, 2811–2821, doi:10.1109/TGRS.2017.2783902.
5. Shao, Z.; Zhou, Z.; Huang, X.; Zhang, Y. MRENet: Simultaneous Extraction of Road Surface and Road Centerline in Complex Urban Scenes from Very High-Resolution Images. *Remote Sens.* **2021**, *13*, 239, doi:10.3390/rs13020239.
6. Zhang, R.; Shao, Z.; Huang, X.; Wang, J.; Li, D. Object Detection in UAV Images via Global Density Fused Convolutional Network. *Remote Sens.* **2020**, *12*, 3140, doi:10.3390/rs12193140.
7. Yao, H.; Liu, Y.; Wei, Y.; Tang, X.; Li, Z. Learning from Multiple Cities: A Meta-Learning Approach for Spatial-Temporal Prediction. In *Proceedings of the The World Wide Web Conference on-WWW '19*; ACM Press: San Francisco, CA, USA, 2019; pp. 2181–2191.
8. Li, W.; Wang, Z.; Wang, Y.; Wu, J.; Wang, J.; Jia, Y.; Gui, G. Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1986–1995, doi:10.1109/JSTARS.2020.2988477.
9. Huang, Z.; Dumitru, C.O.; Pan, Z.; Lei, B.; Datcu, M. Classification of Large-Scale High-Resolution SAR Images With Deep Transfer Learning. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 107–111, doi:10.1109/LGRS.2020.2965558.
10. Pires de Lima, R.; Marfurt, K. Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis. *Remote Sens.* **2020**, *12*, 86, doi:10.3390/rs12010086.
11. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the Proceedings of the 34th International Conference on Machine Learning*; Precup, D., Teh, Y.W., Eds.; PMLR: Sydney, NSW, Australia, 2017; Volume 70, pp. 1126–1135.
12. Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; Huang, J.-B. A Closer Look at Few-Shot Classification. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
13. Wang, Y.; Yao, Q.; Kwok, J.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.* **2020**, *53*, 1–34.

14. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in Neural Networks: A Survey. *arXiv* **2020,** arXiv:2004.05439.
15. Raghu, A.; Raghu, M.; Bengio, S.; Vinyals, O. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
16. Hochreiter, S.; Younger, A.S.; Conwell, P.R. Learning to Learn Using Gradient Descent. In *Artificial Neural Networks—ICANN 2001*; Dorffner, G., Bischof, H., Hornik, K., Eds.; Lecture Notes in Computer Science; Springer: Berlin, Heidelberg, 2001; Volume 2130, pp. 87–94, ISBN 978-3-540-42486-4.
17. Li, H.; Cui, Z.; Zhu, Z.; Chen, L.; Zhu, J.; Huang, H.; Tao, C. RS-MetaNet: Deep Metametric Learning for Few-Shot Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–12, doi:10.1109/TGRS.2020.3027387.
18. Ruswurm, M.; Wang, S.; Korner, M.; Lobell, D. Meta-Learning for Few-Shot Land Cover Classification. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–16 June 2020*; IEEE: Piscataway Township, NJ, USA; pp. 788–796.
19. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359, doi:10.1109/TKDE.2009.191.
20. Erhan, D.; Courville, A.; Bengio, Y.; Vincent, P. Why Does Unsupervised Pre-Training Help Deep Learning? In *Proceedings of the Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*; Teh, Y.W., Titterington, M., Eds.; PMLR: Sardinia, Italy, 2010; Volume 9, pp. 201–208.
21. Saikia, T.; Brox, T.; Schmid, C. Optimized Generic Feature Learning for Few-Shot Classification across Domains. *arXiv* **2020**, arXiv:2001.07926.
22. Chen, Z.; Zhang, T.; Ouyang, C. End-to-End Airplane Detection Using Transfer Learning in Remote Sensing Images. *Remote Sens.* **2018**, *10*, doi:10.3390/rs10010139.
23. Li, X.; Zhang, L.; Du, B.; Zhang, L.; Shi, Q. Iterative Reweighting Heterogeneous Transfer Learning Framework for Supervised Remote Sensing Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 2022–2035, doi:10.1109/JSTARS.2016.2646138.
24. Chen, X.; Xie, S.; He, K. An Empirical Study of Training Self-Supervised Visual Transformers. *arXiv* **2021**, arXiv:2104.02057.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Proceedings of the Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
26. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
27. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
29. Srinivas, A.; Lin, T.-Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck Transformers for Visual Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 16519–16529.
30. Ravi, S.; Larochelle, H. Optimization as a model for few-shot learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; p. 11.
31. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.B.; Larochelle, H.; Zemel, R.S. Meta-Learning for Semi-Supervised Few-Shot Classification. *arXiv* **2018**, arXiv:1803.00676.
32. Dhillon, G.S.; Chaudhari, P.; Ravichandran, A.; Soatto, S. A Baseline for Few-Shot Image Classification. *arXiv* **2020**, arXiv:1909.02729.
33. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33rd International Conference on Machine Learning*; Balcan, M.F., Weinberger, K.Q., Eds.; PMLR: New York, NY, USA, 20 June 2016; Volume 48, pp. 1842–1850.
34. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to Compare: Relation Network for Few-Shot Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
35. Snell, J.; Swersky, K.; Zemel, R. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 4077–4087.
36. Antoniou, A.; Storkey, A.J. Learning to Learn by Self-Critique. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., dAlché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
37. Rusu, A.A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; Hadsell, R. Meta-Learning with Latent Embedding Optimization. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
38. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv* **2017**, arXiv:1707.09835.
39. Keskar, N.S.; Socher, R. Improving Generalization Performance by Switching from Adam to SGD. *arXiv* **2017**, arXiv:1712.07628.

40. Metz, L.; Maheswaranathan, N.; Cheung, B.; Sohl-Dickstein, J. Meta-Learning Update Rules for Unsupervised Representation Learning. *arXiv* **2019**, arXiv:1804.00222.
41. Schmitt, M.; Hughes, L.H.; Qiu, C.; Zhu, X.X. SEN12MS—A Curated Dataset of Georeferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion. *arXiv* **2019**, arXiv:1906.07789.