remote sensing

MDPI

Article

Dictionary Learning for Few-Shot Remote Sensing Scene Classification

Yuteng Ma [1,2,3], Junmin Meng [1,3,*], Baodi Liu [4], Lina Sun [1,3], Hao Zhang [1,2,3] and Peng Ren [3]

[1] First Institute of Oceanography, Ministry of Natural Resources, Qingdao 266061, China
[2] College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China
[3] Technology Innovation Center for Ocean Telemetry, Ministry of Natural Resources, Qingdao 266061, China
[4] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China
* Correspondence: mengjm@fio.org.cn

Abstract: With deep learning-based methods growing (even with scarce data in some fields), few-shot remote sensing scene classification (FSRSSC) has received a lot of attention. One mainstream approach uses base data to train a feature extractor (FE) in the pre-training phase and employs novel data to design the classifier and complete the classification task in the meta-test phase. Due to the scarcity of remote sensing data, obtaining a suitable feature extractor for remote sensing data and designing a robust classifier have become two major challenges. In this paper, we propose a novel dictionary learning (DL) algorithm for few-shot remote sensing scene classification to address these two difficulties. First, we use natural image datasets with sufficient data to obtain a pre-trained feature extractor. We fine-tune the parameters with the remote sensing dataset to make the feature extractor suitable for remote sensing data. Second, we design the kernel space classifier to map the features to a high-dimensional space and embed the label information into the dictionary learning to improve the discrimination of features for classification. Extensive experiments on four popular remote sensing scene classification datasets demonstrate the effectiveness of our proposed dictionary learning method.

Keywords: remote sensing scene; dictionary learning; few-shot image classification

Citation: Ma, Y.; Meng, J.; Liu, B.; Sun, L.; Zhang, H.; Ren, P. Dictionary Learning for Few-Shot Remote Sensing Scene Classification. *Remote Sens.* **2023**, *15*, 773. https://doi.org/10.3390/rs15030773

Academic Editor: Johannes R. Sveinsson

Received: 7 December 2022
Revised: 12 January 2023
Accepted: 21 January 2023
Published: 29 January 2023

1. Introduction

Remote sensing is an advanced and practical comprehensive type of observation technology. It obtains ground object information through observation at high altitudes and systematically analyzes it. Remote sensing scene classification (RSSC) is widely used in resource investigation [1], urban planning [2], land use and cover [3], and environmental monitoring [4]. Deep learning techniques, particularly convolutional neural networks (CNNs) [5], have gained popularity in recent years, and they are now the most advanced remote sensing image classification solutions available [6–9]. However, deep learning-based methods are unable to be incompetent without large-labeled data; the extreme lack of data and the high cost of data acquisition limit the application of deep learning-based models in remote sensing. Few-shot learning (FSL) [10–12], which has recently gained popularity in place of conventional classification methods, attempts to develop a model that can swiftly learn new concepts from a small number of labeled samples. In this paper, few-shot learning was 'demonstrated' to RSSC to solve the problem of insufficient labeled data and improve classification efficiency.

Researchers usually study the few-shot remote sensing scene classification in the decoupling mode. Specifically, the classification task includes two stages. (1) The pre-training stage employs base data to generate a feature extractor based on CNN. (2) The meta-test stage utilizes the trained feature extractor to extract the features of the novel data

and then designs a classifier to recognize the unlabeled samples. Compared with traditional few-shot image classification, remote sensing data are more scarce; thus, few-shot remote sensing scene classification faces two severe challenges.

The first challenge we need to address is how to obtain a feature extractor suitable for remote sensing images. To deal with the few-shot natural image classification task, researchers propose using pre-training or fine-tuning strategies to improve the feature extraction ability of the model, which requires the use of a large amount of base data in the pre-training stage to generate a CNN-based feature extractor. However, due to the scarcity of data in remote sensing datasets, using remote sensing data directly for model training leads to an overfitting problem. To address this problem, we employ another few-shot natural image dataset with many data (e.g., tiered-ImageNet) to generate a feature extractor during the pre-training stage. However, due to the large data gap between the remote sensing dataset and the few-shot natural image dataset, there is a large domain transfer, leading to the feature extractor based on the dataset of few-shot images not adapting to remote sensing data. Therefore, we fine-tune the parameters of some layers of the network with the remote sensing dataset of interest to make the feature extractor suitable for remote sensing data classification. In addition, we design a feature extractor by combining self-supervised rotation loss with classification loss to enhance the generalization performance of extracted features.

The second challenge involves designing a robust classifier with a few labeled samples in the meta-test stage. Since there is a domain transfer between the novel data and the base data, the features of the novel data extracted from the feature extractor lack discriminative information, which is called the "negative transfer" problem. In addition, due to the spatial resolution limitations of remote sensing images, there may be noise interference in the features, which leads to the failure of traditional classifiers to effectively classify. We illustrate the t-SNE visualization of features in Figure 1. As shown in Figure 1, different categories are more dispersed, while feature embeddings within the same category are more concentrated. However, a traditional linear classifier cannot adequately distinguish among feature embeddings of test samples that cross each other in spatial distributions.

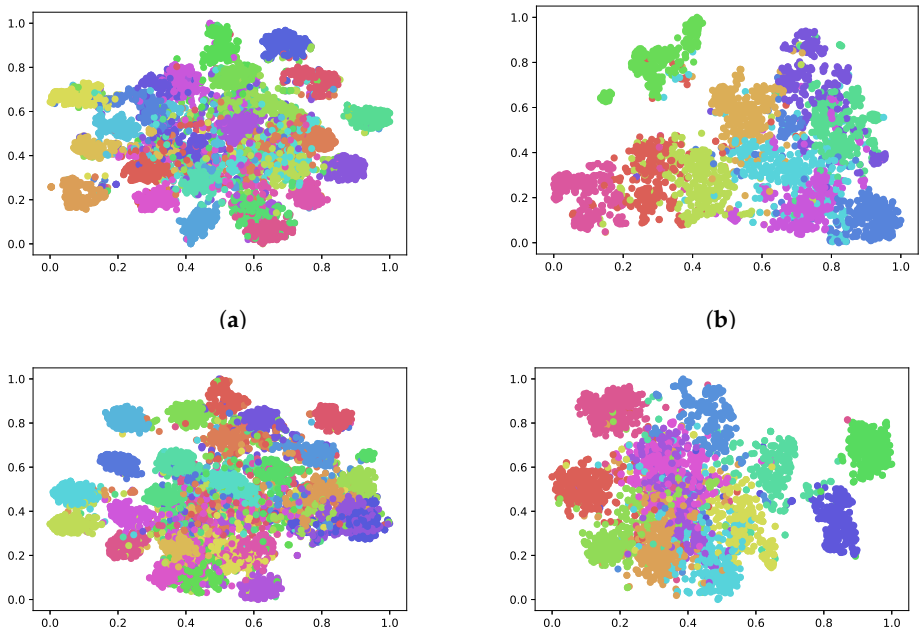


Figure 1. The t-distributed stochastic neighbor embedding (t-SNE) visualization; (**a**,**b**) indicate the training features and the test features of the NWPU-RESISC45 dataset, respectively; (**c**,**d**) represent the training features and test features of the RSD46-WHU dataset, respectively.

In this paper, we propose a novel dictionary-learning algorithm for few-shot remote sensing scene classification to attack the existing challenges. First, we adopt a novel pre-training combined with fine-tuning strategy. We first train a feature extractor with the few-shot natural image dataset and then fine-tune the parameters of some layers of the network with the remote sensing dataset of interest to make the feature extractor suitable for remote sensing data classification. The pre-training model extracts the prior knowledge of the training set, and this high-order model can be used for feature extraction of various tasks. In the process of fine-tuning the model, the last layers are altered to encode specific features of remote sensing data, while the earlier layers are kept since they encode more general features. In addition, to enhance the generalization performance of extracted features, we design a feature extractor by combining self-supervised rotation loss with classification loss. Then, we suggest using a kernel space classifier instead of the traditional linear classifier to map the sample features into high-dimensional kernel space to solve the problem of linear inseparability. To solve the "negative transfer" problem, we propose a dual form of dictionary learning and embed label information into dictionary learning, which improves the discrimination of features. The framework of the method is shown in Figure 2.

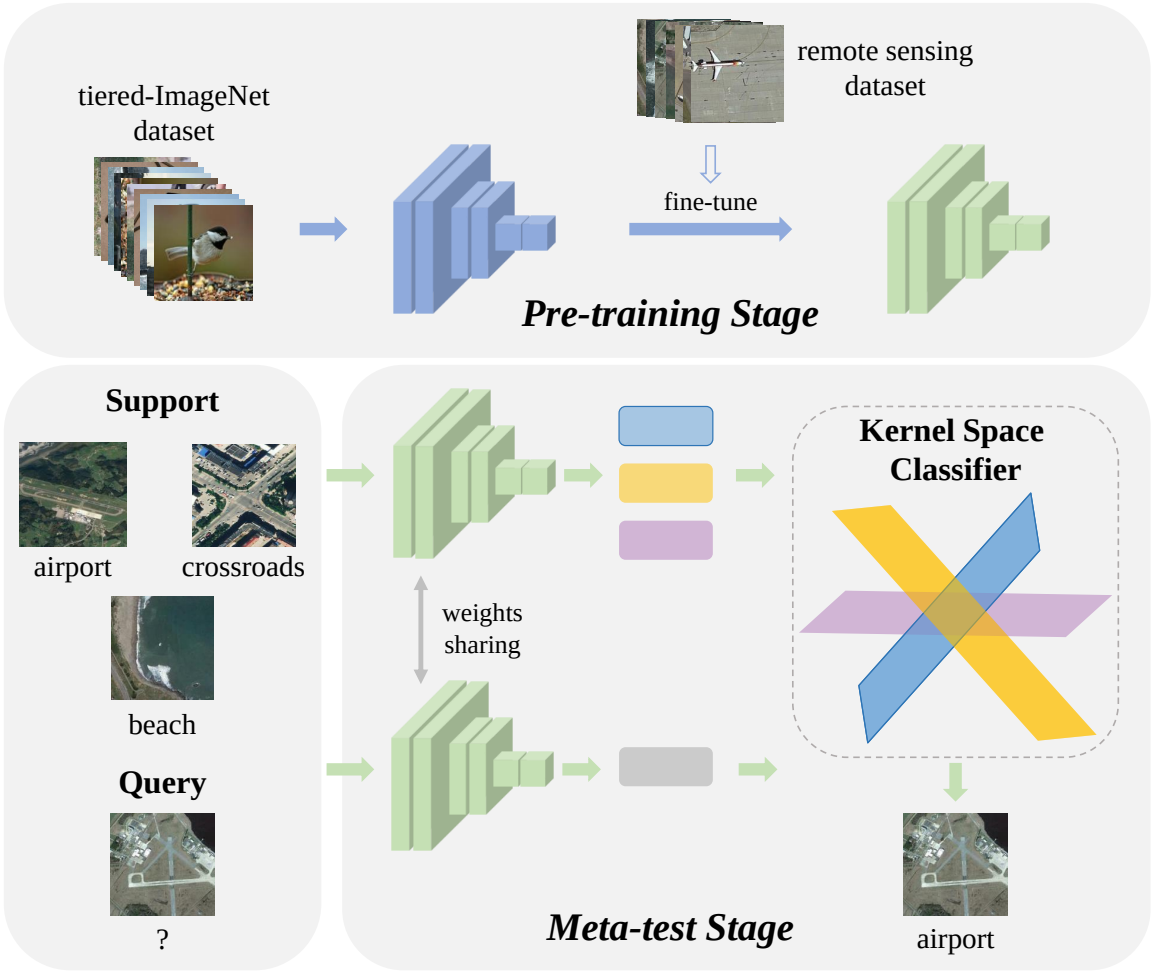


Figure 2. The framework of the proposed DL method consists of two stages: (1) the pre-training stage uses the tiered-ImageNet dataset to train a feature extractor, and then employs a remote sensing dataset to fine-tune the extractor. (2) The meta-test stage inputs the support set and the query set samples into the trained feature extractor to obtain the embedding features. We use the support features to train the kernel space classifier and predict the category of the query sample.

The main contributions of this paper are as follows.

- We designed a kernel space classifier for the few-shot remote sensing scene classification task, which introduces the kernel space into dictionary learning and improves the classification performance.
- We propose a dual form of dictionary learning and embed label information into dictionary learning, improving feature discrimination. Further experiments show that the proposed method can effectively solve the problem of "negative transfer".
- The proposed method was evaluated on four remote sensing datasets—NWPU-RESISC45, RSD46-WHU, UC Merced, and WHU-RS19. It demonstrated satisfactory performance compared with the state-of-the-art methods.

2. Related Work

2.1. Remote Sensing Scene Classification

Remote sensing scene classification is a research hotspot. Existing RSSC methods can be divided into three categories: (1) Low-level feature descriptors. These methods distinguish remote sensing scenes by low-level visual features, such as spectrum, color, texture, and structure. Local descriptors, e.g., the histograms of oriented gradients (HOG) [13], scale-invariant feature transform (SIFT) [14], and local binary patterns (LBPs) [15], are widely used in modeling local changes of structures in remote sensing images due to their invariability to geometric and photometric transformations. (2) Mid-level visual representations. These methods combine extracted local visual features with higher-order statistical patterns to develop a holistic scene representation. Because of their simplicity and efficiency, the bag-of-visual-words (BovW) model [16] and its variants have been widely used in RSSC. (3) High-level vision information. These methods rely on deep neural networks. They adopt multi-stage global feature-learning structures to learn image features adaptively and classify remote sensing scenes as end-to-end problems, further improving the classification performance by multi-feature fusion, multi-model fusion [17], and multi-decision fusion. Compared with low-level and mid-level methods, the methods based on deep learning have become the most advanced solutions in the field of RSSC because they can learn more abstract and discriminative semantic features [18–20].

2.2. Few-Shot Remote Sensing Scene Classification

In recent years, due to the widespread interest in few-shot learning, researchers have applied few-shot learning to RSSC to deal with the problem of scarce remote sensing data. Two main approaches to few-shot learning based on remote sensing are (1) optimization-based methods. These methods consider the fact that it is challenging for ordinary gradient descent methods to fit in few-shot scenarios, so the task of few-shot classification is accomplished by adjusting the optimization method. MAML [21], reptile [22], and LEO [23] are all FSL methods based on optimization. Dalal, A. A. et al. [24] utilized MAML for the RSSC task and achieved satisfactory results With little remote sensing data. (2) Metric-based methods. The principle of these methods is to build task-specific distance metrology functions independently through different tasks. The Siamese Network [25], matching networks [26], prototypical networks [10], and relation networks [27] are all FSL methods based on the metric. Li, L. et al. [28] suggested a matching network-based method for FSRSSC. The prototypical network was used for RSSC by Alajaji, D., Zhang, P. et al. [29,30], and achieved excellent performance.

2.3. Negative Transfer

Pre-trained feature extraction models are essential toward improving few-shot classification performance. Due to the scarcity of remote sensing data, RSSC tasks usually utilize pre-trained models for transfer learning [29]. However, the pre-trained models do not adapt well to remote sensing data due to large data gaps. We call this a "negative transfer" problem, which seriously affects classification performance. To address the problem, a learning-based few-shot approach was proposed by Dvornik, N. et al. [31]. It integrates

multiple pre-trained models and calculates the final result by voting or averaging based on the output of the models. In order to eliminate the confounding, Yue, Z. et al. [32] provided a causality-based explanation of the causes of confounding introduced in pre-trained knowledge and carried out feature-based and class-based modifications. Shao, S. et al. [33] employed the subspace approach to project the multi-head features into a uniform space to acquire low-dimensional representation. The negative transfer problem is somewhat mitigated by integrating the derived principal component features to produce more discriminative information.

3. Problem Setup

3.1. Problem Definition

In this paper, the few-shot remote sensing scene classification task contains two stages: the pre-training stage and the meta-test stage.

In the pre-training stage, we employ base data $D_{base} = \{(x_i, y_i)\}_{i=1}^{N}$ to train a feature extractor based on CNN, where x_i indicates the i_{th} sample, y_i denotes the corresponding label, and N represents the total number of base data. Then we fix the parameters of the feature extractor.

In the meta-test stage, we use the trained feature extractor to extract the features of the novel data, and then design a classifier to recognize the test samples. The novel data $D_{novel} = \{T_i\}_{i=1}^{M}$ consists of a series of meta-tasks, where T_i represents the i_{th} meta-task and M is the sum of novel data. Notably, the set of categories in D_{base} and D_{novel} are disjoint. Each meta-task contains support set S and query set Q. Specifically, $S = \{(x_i, y_i)|i = 1, 2, \cdots, C \times N_s\}$ represents the support set, where C denotes the class number, N_s represents the number of samples for each class. $Q = \{(x_i, y_i)|i = 1, 2, \cdots, C \times N_q\}$ is the query set, where N_q denotes the sample number in each class. In this paper, we define C as 5, and N_s as 1 or 5.

3.2. Kernel Space Classifier

In prior works, linear classifiers were used to complete classification tasks, such as ICI [12] and MetaOptNet [34]. However, by analyzing meta-training and meta-test data t-SNE visualizations, we found that the novel feature is not highly discriminative and linearly indivisible due to the "negative transfer" problem. If we adopt the linear classifier directly, the performance will not be well. We adopt two strategies to improve existing problems. Firstly, in order to improve the discriminability of features, we introduce feature reconstruction errors based on the linear classifier to map sample features to a more discriminative space, which can alleviate the problem of "negative transfer". At the same time, the linear indivisible features of low dimensional space are often mapped to the higher dimensional space, which will become linearly divisible. Therefore, we introduce the dual form of dictionary learning to complete the classification of new class samples in the kernel space, which can increase the linear separability of new class samples.

In this paper, we use a kernel space classifier; we map the sample feature space to the high-dimensional kernel space, then carry out the classification task. Denote that the training sample matrix $X = [X_1, X_2, \ldots, X_C] \in \mathbb{R}^{D \times N}$, where N is the number of training samples, D represents the dimension of sample features, C is the number of training sample categories, and X_c denotes the training sample features of class c. Supposing that $Y \in \mathbb{R}^{C \times N}$ is the corresponding one-hot label matrix. We define a feature mapping function ϕ: $\mathbb{R}^D \to \mathbb{R}^E$ ($D \ll E$), which transforms the initial feature space into a high-dimensional kernel space: $X \to \phi(X)$.

We combine the reconstruction and classification errors to form a unified objective function and introduce the ℓ-norm regularization term to enhance the sparsity. The objective function is defined as follows:

$$\begin{aligned} &\arg\min_{W,V,S} \|\phi(X) - \phi(X)VS\|_H^2 + 2\alpha\|S\|_{l_1} + \eta\|Y - WS\|_F^2 \\ &s.t. \|\phi(X)V_{\bullet k}\|_H \leq 1, \|W_{\bullet k}\|_2 \leq 1, \forall k = 1, 2, \ldots, K. \end{aligned} \tag{1}$$

where $V \in \mathbb{R}^{N\times K}$ and $\phi(X)V$ is the dual form of the dictionary. K is the size of the dictionary. $S \in \mathbb{R}^{K\times N}$ is the corresponding sparse code. W is a classifier learned from the given label matrix Y. $(\bullet)_{\bullet k}$ denotes the k_{th} column vector of the matrix $(\bullet)$. α and η are the regularization parameters to control the trade-off between fitting goodness and sparseness and balance the classification contribution to the overall objective, respectively. Then we optimize the objective function.

(1) When V and S are fixed, Equation (1) can be rewritten as:

$$\begin{aligned} &f(W) = \arg\min_{W} \|Y - WS\|_F^2 \\ &s.t. \|W_{\bullet k}\|_2 \leq 1, \forall k = 1, 2, \ldots, K. \end{aligned} \tag{2}$$

We solve this problem by introducing Lagrangian, and the Equation (2) can be rewritten as:

$$\mathcal{L}(W, \gamma_k) = \|Y - WS\|_H^2 + \sum_{k=1}^{K} \lambda_k (1 - \|W_{\bullet k}\|_F) \tag{3}$$

Let $\frac{\partial L(W)}{\partial W_{\bullet k}} = 0$, $W_{\bullet k}$ can be obtained as:

$$W_{\bullet k} = \frac{YS_{k\bullet}^T - \tilde{W}^k SS_{k\bullet}^T}{\left\|YS_{k\bullet}^T - \tilde{W}^k SS_{k\bullet}^T\right\|_2} \tag{4}$$

where $\tilde{W}^k = \begin{cases} W_{\bullet p}, & p \neq k \\ 0, & p = k \end{cases}$, $(\bullet)_{k\bullet}$ is the k_{th} row vector of matrix $(\bullet)$.

(2) When W and S are fixed, Equation (1) can be rewritten as:

$$\begin{aligned} &f(V) = \|\phi(X) - \phi(X)VS\|_H^2 \\ &s.t. \|\phi(X)V_{\bullet k}\|_H \leq 1, \forall k = 1, 2, \ldots, K. \end{aligned} \tag{5}$$

We solve this problem by introducing Lagrangian, and Equation (5) can be rewritten as:

$$\begin{aligned} \mathcal{L}(V, \lambda_k) = &\|\phi(X) - \phi(X)VS\|_H^2 \\ &+ \sum_{k=1}^{K} \lambda_k (1 - \|\phi(X)V_{\bullet k}\|_F) \end{aligned} \tag{6}$$

Let $\frac{\partial \mathcal{L}(V,\lambda_k)}{\partial V_{\bullet k}} = 0$, we obtain the solution of $V_{\bullet k}$ as:

$$V_{\bullet k} = \frac{S_{k\bullet}^T - [\tilde{V}^k SS^T]_{\bullet k}}{[SS^T]_{kk} - \lambda_k} \tag{7}$$

where $\tilde{V}^k = \begin{cases} V_{\bullet p}, & p \neq k \\ 0, & p = k \end{cases}$. Substituting $V_{\bullet k}$ into Equation (6), and only keeping the term, including $V_{\bullet k}$, we obtain:

$$\begin{aligned} \mathcal{L}(V, \lambda_k) = &\frac{(S_{k\bullet}^T - [\tilde{V}^k SS^T]_{\bullet k})^T k(X, X)(S_{k\bullet}^T - [\tilde{V}^k SS^T]_{\bullet k})}{\lambda_k - [SS^T]_{kk}} \\ &+ \lambda_k \end{aligned} \tag{8}$$

where $k(X, X) = \phi(X)^T \phi(X)$ represents the kernel function. Then, we obtain λ_k and substitute it into $V_{\bullet k}$,

$$V_{\bullet k} = \frac{S_{k\bullet}^T - [\tilde{V}^k S S^T]_{\bullet k}}{\pm\sqrt{(S_{k\bullet}^T - [\tilde{V}^k S S^T]_{\bullet k})^T k(X,X)(S_{k\bullet}^T - [\tilde{V}^k S S^T]_{\bullet k})}} \tag{9}$$

We obtain two solutions with $\pm$ signs from Equation (9). The sign of $V_{\bullet k}$ is not vital because it can be easily absorbed by converting between $S_{\bullet k}$ and $-S_{\bullet k}$.

(3) When W and V are fixed, we introduce an auxiliary variable Z, and the Equation (1) can be rewritten as:

$$\begin{aligned} f(S, Z, L) = &\|\phi(X) - \phi(X)VS\|_H^2 + \eta\|Y - WS\|_F^2 \\ &+ 2\alpha\|Z\|_{l_1} + L^T(S - Z) + \rho\|S - Z\|_F^2 \end{aligned} \tag{10}$$

where $\rho > 0$ is the penalty parameter, and $L = [l_1, l_2, ..., l_N] \in \mathbb{R}^{K\times N}$ is the augmented Lagrange multiplier. After fixing W and V, we initialize the S_0, Z_0, and L_0 to be zero matrices.

We fix L and Z and update S. Equation (10) can be rewritten as follows:

$$\begin{aligned} f(S) = &\|\phi(X) - \phi(X)VS\|_H^2 + \eta\|Y - WS\|_F^2 \\ &+ L^T(S - Z) + \rho\|S - Z\|_F^2 \end{aligned} \tag{11}$$

Let $\frac{\partial f}{\partial S} = 0$, the closed-form solution of S is:

$$\begin{aligned} S_{m+1} = &[V^T k(X,X)V + \eta W_m^T W_m + \rho I]^{-1} \\ &\times [V^T k(X,X) + \eta W_m^T Y + \rho Z_m - L_m] \end{aligned} \tag{12}$$

where $m(m = 0, 1, 2, \dots)$ denotes the iteration number and $(\bullet)_m$ means the value of matrix $(\bullet)$ after m_{th} iteration.

We fix S and L and update Z. Equation (10) can be rewritten as follows:

$$f(Z) = +2\alpha\|Z\|_{l_1} + L^T(S - Z) + \rho\|S - Z\|_F^2 \tag{13}$$

The closed-form solution of Z is:

$$\begin{aligned} Z_{m+1} = &max\{S_{m+1} + \frac{L_m}{\rho} - \frac{\alpha}{\rho}I, 0\} \\ &+ min\{S_{m+1} + \frac{L_m}{\rho} + \frac{\alpha}{\rho}I, 0\} \end{aligned} \tag{14}$$

where I is the identity matrix and 0 is the zero matrix.

We fix S and Z and update the Lagrange multiplier L by the gradient descent method:

$$L_{m+1} = L_m - \theta(S_{m+1} - Z_{m+1}) \tag{15}$$

In the testing stage, the constraint terms are based on l_1-norm sparse constraint. Given the test sample feature $x_t \in \mathbb{R}^{D\times 1}$. After mapping it to kernel space, we exploit the learned dictionary for fitting to obtain the sparse codes s_t. Then, we use the trained classifier W to predict the label of x_t by calculating $max\{Ws_t\}$. The procedure of the proposed dictionary learning method is shown in Algorithm 1.

Algorithm 1 Dictionary learning

Input: $X \in \mathbb{R}^{D\times N}$, $Y \in \mathbb{R}^{C\times N}$, α, η, θ, K
Output: $S \in \mathbb{R}^{K\times N}$, $W \in \mathbb{R}^{C\times K}$, $V \in \mathbb{R}^{N\times K}$
Mapping features $X \in \mathbb{R}^{D\times N}$ to the kernel space $\phi(X) \in \mathbb{R}^{E\times N}$
Initial Z_0, L_0, $S_0 \leftarrow$ zeros(K, N)
Initial $W_0 \leftarrow$ rand(C, K), $V_0 \leftarrow$ rand(N, K)
$W_{\bullet k} = \frac{W_{\bullet k}}{\|W_{\bullet k}\|_2}$, $V_{\bullet k} = \frac{V_{\bullet k}}{\|V_{\bullet k}\|_2}$, $(k = 1, 2, ..., K)$
for m = 0 to maxiter **do**
    Using Equation (12) to update S_{m+1}
    Using Equation (14) to update Z_{m+1}
    Using Equation (15) to update L_{m+1}
    for k = 0 to K **do**
        Using Equation (4) to update $(W_{\bullet k})_{m+1}$
        Using Equation (9) to update $(V_{\bullet k})_{m+1}$
    end for
end for
return S, W, V

4. Experiments and Results

4.1. Datasets

In this paper, we employ the tiered-ImageNet [35] dataset to obtain the pre-trained model, and evaluate the proposed DL method on four datasets of remote sensing images, including NWPU-RESISC45 [36], RSD46-WHU [37,38], UC Merced [16], and WHU-RS19 [39,40]. Figure 3 demonstrates some scenes of the few-shot remote sensing scene classification datasets and the details of the datasets are as follows:

The tiered-ImageNet dataset is a subset of the ILSVRC-12 dataset, which contains 608 classes and each class has 600 images with a size of 84 × 84. We divide tiered-ImageNet into 3 sections. Specifically, the base set contains 351 classes for meta-training, the validation set contains 97 classes for meta-validation, and the novel set contains 160 classes for meta-test.

The NWPU-RESISC45 dataset is proposed by Cheng et al. [36], and 45 classes with 700 remote sensing scene images in each class consist of it. The dimension of each of these samples is 256 × 256 pixels. We follow the split introduced in prior work [30], to divide NWPU-RESISC45 into 25 classes for meta-training, 8 classes for meta-validation, and 12 classes for meta-test.

The RSD46-WHU dataset comes from Tianditu and Google Earth and contains 46 classes with 428–3000 images per class, for a total of 117,000 images. The ground resolution of most classes is 0.5 m, and the others are about 2 m. We split the 46 classes into 26, 8, and 12 classes for meta-training, meta-validation, and meta-test.

The UC Merced dataset includes a total of 2,100 images from 21 scenarios, each containing 100 images 256 × 256 pixels in size. The original image was downloaded by the USGS from national maps in various parts of the country. Based on the division of previous work [28], the UC Merced dataset is divided into 10, 6, and 5 for meta-training, meta-validation, and meta-test, respectively.

The WHU-RS19 is a dataset of satellite images exported from Google Earth, which contains 1005 images divided into 19 classes of scenes in high-resolution satellite imagery. Each class has about 50 images of 600-pixel size. Following the split setting introduced in prior work [28], we divided it into 9 classes for meta-training, 6 classes for meta-validation, and 5 classes for meta-test. The details of the datasets are shown in Table 1.

Table 1. Category information of datasets.

Dataset	Pre-Training	Meta-Validation	Meta-Test
tiered-ImageNet	351	97	160
NWPU-RESISC45	25	8	12
RSD46-WHU	26	8	12
UC Merced	10	6	5
WHU-RS19	9	6	5

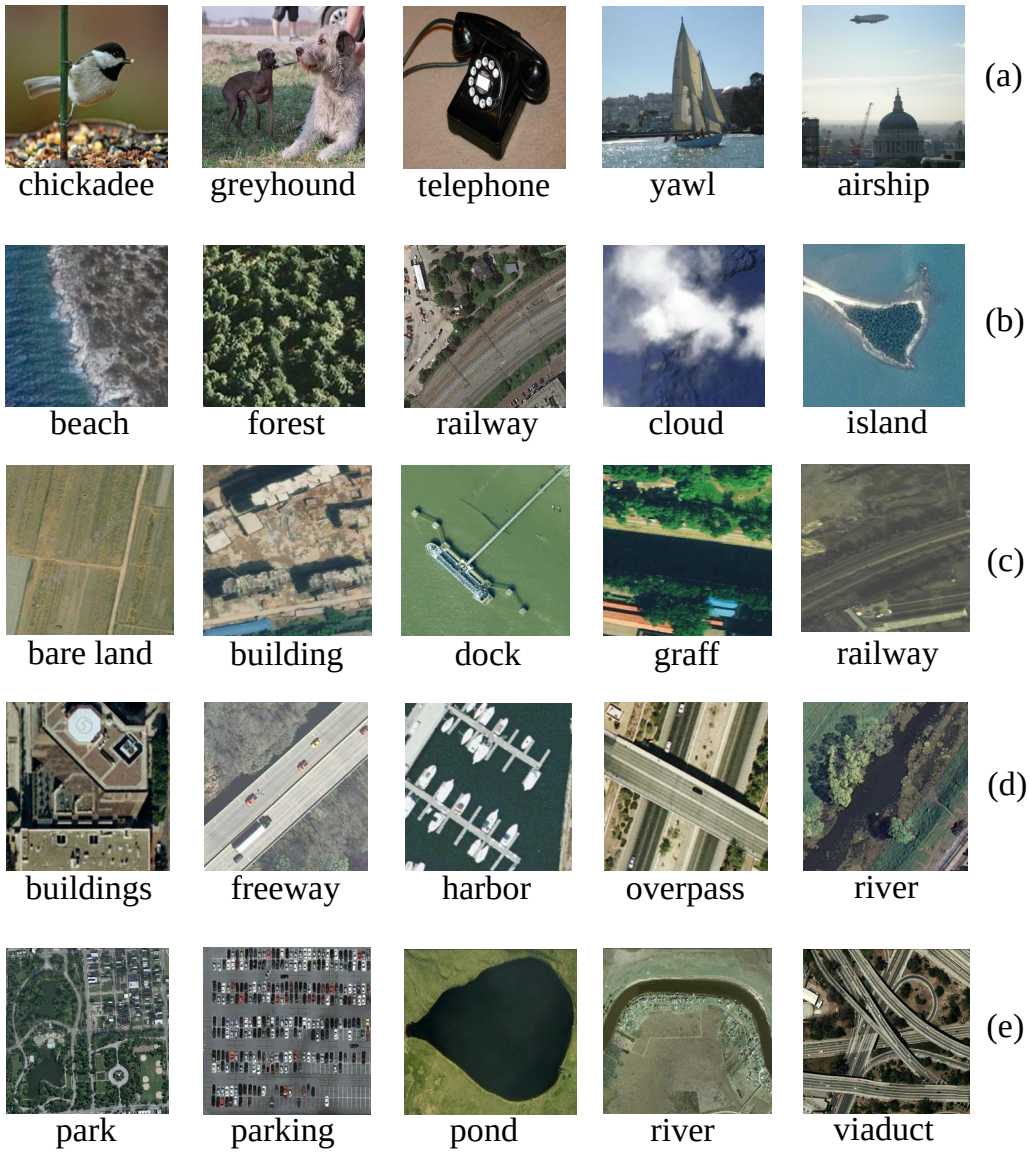


Figure 3. Example samples of the five datasets used in this paper. (**a**) tiered-ImageNet, (**b**) NWPU-RESISC45, (**c**) RSD46-WHU, (**d**) UC Merced, (**e**) WHU-RS19.

4.2. Implementation Details

The implementation details are presented in this section. We implemented our method using the deep learning framework PyTorch 1.1.0 and completed the experiments with a Tesla-V100 GPU (16G memory). In the pre-training stage, we combine the classification loss with the self-supervised rotation loss to build a feature extractor based on the ResNet-12 network as shown in Figure 4. We used a few-shot natural image dataset (tiered-ImageNet) to train a model as the pre-trained feature extractor. We employ stochastic gradient descent

(SGD) as an optimizer, with a weight decay of 10^{-4} and a Nesterov momentum of 0.9. The initial learning rate is set at 0.1, which is later reduced to 0.01 at the 30-epoch mark, 0.001 at the 60-epoch mark, and 0.0001 at the 90-epoch mark. To fine-tune the pre-training model, we select the remote sensing dataset corresponding to the classification task. We initialized the learning rate to 0.1 and trained the model with 20 epochs. Additionally, standard data augmentation techniques, such as color dithering, random clipping, and horizontal flipping are applied during the pre-processing data stage.

In the meta-test stage, we use the kernel space classifier to complete the classification. In Equation (1), we fix parameter α as 0.15 and η as 1.0. In Equation (15), the gradient descent parameter θ is 0.5. We use three different kernels: linear kernel ($k(x,y) = x^T y$), poly kernel ($k(x,y) = (1 + x^T y)^p$), and RBF kernel ($k(x,y) = exp(-\gamma \|x - y\|^2)$). Here, we set $p = 1$ and $\gamma = 2$. Following the FSL experimental setting, the performance is evaluated in the 5-way 5-shot case or 5-way 1-shot case with 15 query samples.

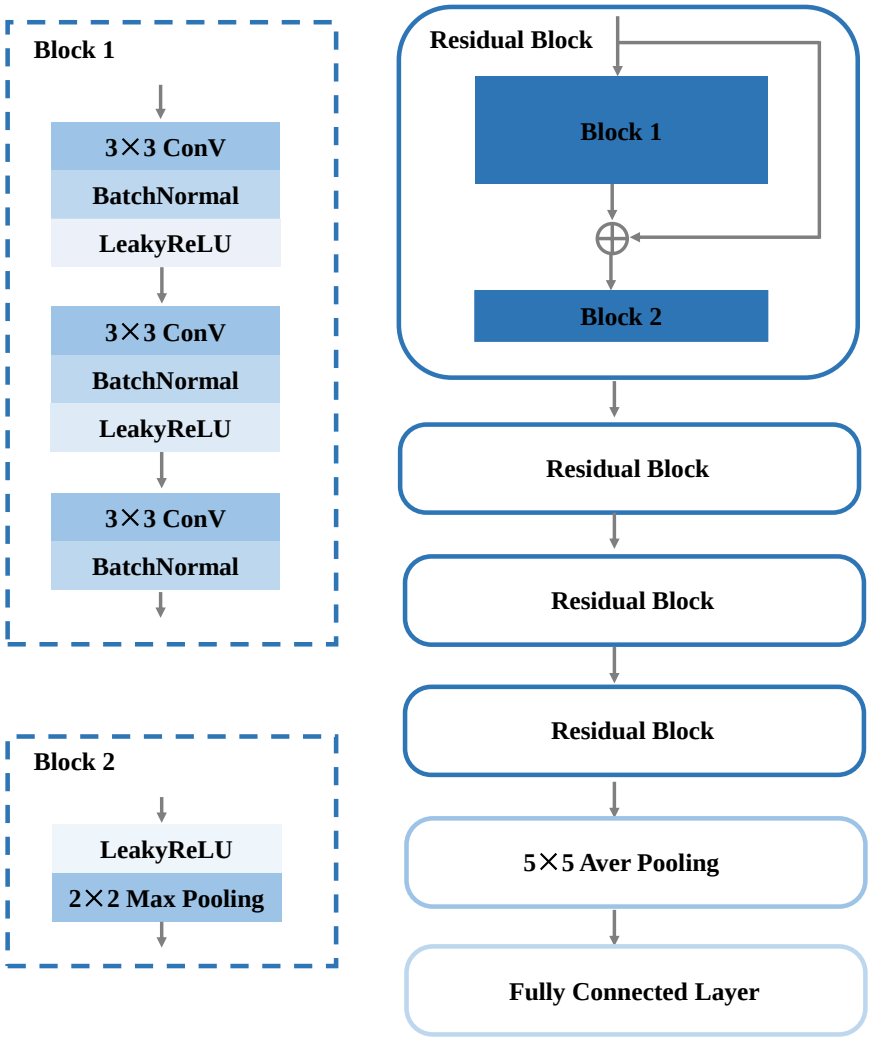


Figure 4. Schematic diagram of ResNet-12.

4.3. Experimental Results

The proposed method is compared with several state-of-the-art methods. The experimental results, which are shown in Tables 2–5, demonstrate that our DL method performs better than others on four remote sensing scene image datasets.

Table 2. The few-shot classification accuracies with 95% confidence intervals over 600 episodes in the NWPU-RESISC45.

Method	Backbone	5-Way 5-Shot	5-Way 1-Shot
LLSR [41]	ConV4	72.90	51.43
MatchingNet [26]	ConV5	47.10	37.61
DLA-MatchNet [28]	ConV5	81.63 ± 0.46	68.80 ± 0.70
Meta-SGD [42]	ConV5	75.75 ± 0.65	60.63 ± 0.90
ProtoNet [10]	ResNet12	80.19 ± 0.52	62.78 ± 0.85
MAML [21]	ResNet12	72.94 ± 0.63	56.01 ± 0.87
TADAM [43]	ResNet12	82.36 ± 0.54	62.25 ± 0.79
TAE-Net [44]	ResNet12	82.37 ± 0.52	69.13 ± 0.83
D-CNN [45]	ResNet12	53.60 ± 5.34	36.00 ± 6.31
DSN-MR [46]	ResNet12	81.67 ± 0.49	66.93 ± 0.51
MetaOptNet [34]	ResNet12	80.41 ± 0.41	62.72 ± 0.64
TPN [47]	ResNet12	78.50 ± 0.56	66.51 ± 0.87
MetaLearning [30]	ResNet12	84.66 ± 0.12	69.46 ± 0.22
RelationNet [27]	ResNet12	75.78 ± 0.57	55.84 ± 0.88
RS-SSKD [48]	ResNet12	86.26 ± 5.34	70.64 ± 0.22
FEAT [49]	ResNet12	83.51 ± 0.11	68.27 ± 0.19
Ours	ResNet12	88.32 ± 0.43	74.03 ± 0.76

Table 3. The few-shot classification accuracies with 95% confidence over 600 episode intervals in the RSD46-WHU.

Method	Backbone	5-Way 5-Shot	5-Way 1-Shot
RelationNet [27]	ConV4	68.86 ± 0.71	55.18 ± 0.90
ProtoNet [10]	ConV4	69.78 ± 0.73	52.73 ± 0.91
MAML [21]	ConV4	71.95 ± 0.71	52.57 ± 0.89
RelationNet [27]	ResNet12	69.98 ± 0.74	53.73 ± 0.95
MAML [21]	ResNet12	69.28 ± 0.81	54.36 ± 1.04
ProtoNet [10]	ResNet12	77.53 ± 0.73	60.53 ± 0.99
MetaOptNet [34]	ResNet12	82.60 ± 0.46	62.05 ± 0.76
DSN-MR [46]	ResNet12	82.74 ± 0.54	66.53 ± 0.70
D-CNN [45]	ResNet12	58.93 ± 6.14	30.93 ± 7.49
TADAM [43]	ResNet12	82.79 ± 0.54	65.84 ± 0.67
MetaLearning [30]	ResNet12	84.10 ± 0.15	69.08 ± 0.25
RS-SSKD [48]	ResNet12	85.90 ± 0.15	71.73 ± 0.25
FEAT [49]	ResNet12	85.27 ± 0.13	71.04 ± 0.21
Ours	ResNet12	88.19 ± 0.52	74.10 ± 0.88

Table 4. The few-shot classification accuracies with 95% confidence intervals over 600 episodes in the WHU-RS19 dataset.

Method	Backbone	5-Way 5-Shot	5-Way 1-Shot
LLSR [41]	ConV-4	70.65	57.10
ProtoNet [10]	ConV-5	80.70 ± 0.11	58.01 ± 0.16
MatchingNet [26]	ConV-5	54.10	50.13
MAML [21]	ConV-5	62.49 ± 0.51	49.13 ± 0.65
Meta-SGD [42]	ConV-5	61.74 ± 2.02	51.54 ± 2.31
RelationNet [27]	ConV-5	79.75 ± 1.19	60.92 ± 1.86
DLA-MatchNet [28]	ConV-5	79.89 ± 0.33	68.27 ± 1.83
TPN [47]	ResNet-12	71.20 ± 0.55	59.28 ± 0.72
TAE-Net [44]	ResNet-12	88.95 ± 0.53	73.67 ± 0.74
Ours	ResNet-12	93.33 ± 0.23	82.57 ± 0.58

Table 5. The few-shot classification accuracies with 95% confidence over 600 episode intervals in the UC Merced dataset.

Method	Backbone	5-Way 5-Shot	5-Way 1-Shot
MAML [21]	ConV-4	70.80 ± 0.03	51.90 ± 0.05
LLSR [41]	ConV-4	57.40	39.47
RelationNet [27]	ConV-5	61.88 ± 0.50	48.08 ± 1.67
MatchingNet [26]	ConV-5	52.71	34.70
ProtoNet [10]	ConV-5	69.86 ± 0.15	52.27 ± 0.20
DLA-MatchNet [28]	ConV-5	63.01 ± 0.51	53.76 ± 0.62
Meta-SGD [42]	ConV-5	60.82 ± 2.00	50.52 ± 2.61
ProtoNet [29]	SqueezeNet	79.82 ± 0.07	50.45 ± 0.29
TPN [47]	ResNet-12	68.23 ± 0.52	53.36 ± 0.77
MKN [50]	ResNet-12	75.42 ± 0.31	57.29 ± 0.59
MA-deepEMD [51]	ResNet-12	80.39 ± 0.71	61.16 ± 0.31
deepEMD [52]	ResNet-12	77.82 ± 0.66	52.28 ± 0.25
RS-MetaNet [53]	ResNet-12	76.08 ± 0.28	57.23 ± 0.56
TAE-Net [44]	ResNet-12	77.44 ± 0.51	60.21 ± 0.72
Ours	ResNet-12	82.63 ± 0.45	65.44 ± 0.72

In the NWPU-RESISC45 dataset, our method outperforms other methods by at least 3.27% and 2.6% in the cases of 5-way 1-shot and 5-way 5-shot, respectively. In the RSD46-WHU dataset, our method improves by at least 0.77% and 0.47% over other methods in the cases of 5-way 1-shot and 5-way 5-shot. In the UC Merced dataset, our method is at least 3.57% and 2.54% better than other methods in the cases of 5-way 1-shot and 5-way 5-shot. In the WHU-RS19 dataset, our method improved by at least 8.65% and 4.91% in the cases of 5-way 1-shot and 5-way 5-shot.

4.4. Ablation Studies

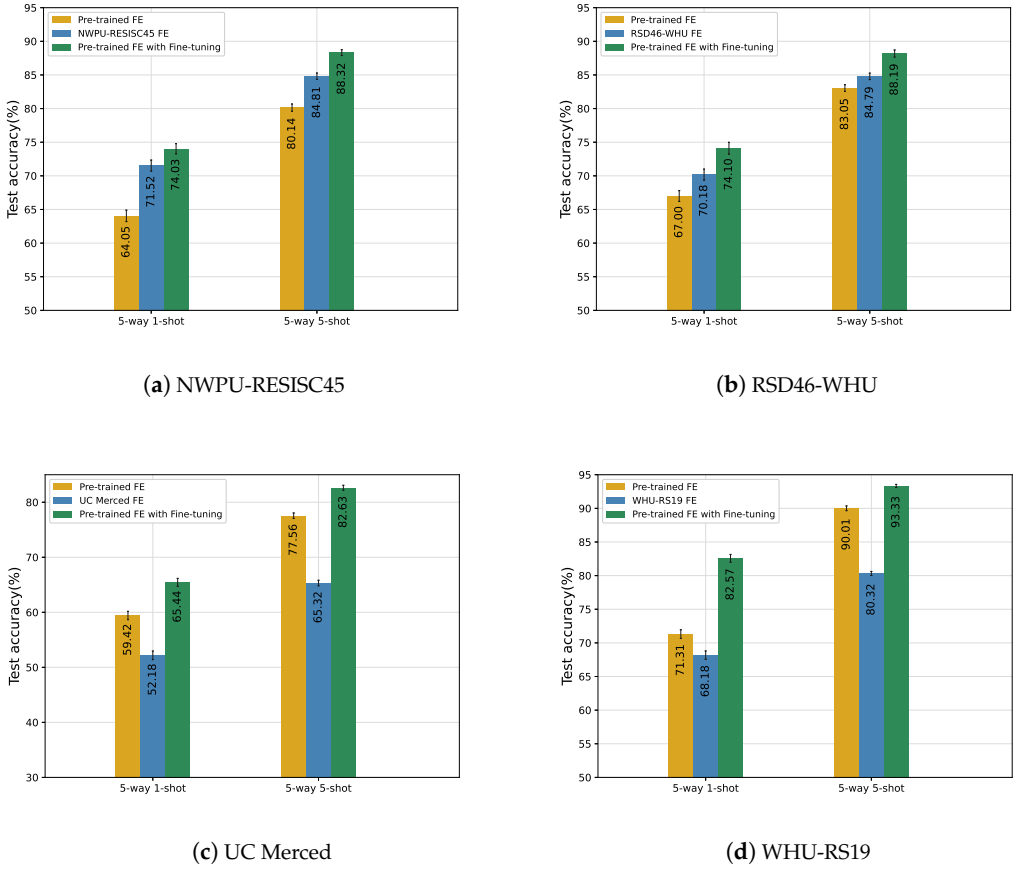
4.4.1. Analysis of Pre-Trained Feature Extractor

In this paper, we first train a feature extractor with a few-shot natural image dataset (e.g., tiered-ImageNet) and then fine-tune the parameters of some layers of the network with the remote sensing dataset of interest to make the feature extractor suitable for remote sensing data classification. To verify the effectiveness of the pre-trained model with fine-tuning strategy, we performed ablation experiments. Figure 5 reports the results in three cases: (1) the pre-trained feature extractor in the tiered-ImageNet dataset; (2) the feature extractor trained with corresponding remote sensing datasets; (3) the pre-trained feature extractor with fine-tuning.

Experimental results show that the pre-trained feature extractor can perform better when labeled remote sensing data are scarce. In the cases of 5-way 1-shot and 5-way 5-shot, the performance of the pre-trained feature extractor is inferior to those of the NWPU-RESISC45 feature extractor and RSD46-WHU feature extractor. However, the performances of the pre-trained feature extractor are 7.24% and 12.24% better than the UC Merced feature extractor in the 1-shot and 5-shot cases, and 3.13% and 9.69% better than the WHU-RS19 feature extractor in 1-shot and 5-shot cases, respectively. This is because the UC Merced and WHU-RS19 datasets contain scarcer labeled samples, making it insufficient to train a feature extractor with superior performance. In contrast, the pre-trained feature extractor is trained with a data-rich tiered-ImageNet dataset, so it is able to extract more generalized features.

In addition, there is significant improvement in using the pre-trained feature extractor with fine-tuning. For the four remote sensing datasets, the results of the pre-trained feature extractor with fine-tuning reach 74.03%, 74.10%, 65.44%, and 82.57% in the 1-shot case, and 88.32%, 88.19%, 82.63%, and 93.33% in the 5-shot case, respectively, performing better than the other two types of feature extractor models. This is due to the corresponding remote

sensing data being used to fine-tune the model parameters so that the feature extractor can be suitable for remote sensing scene classification tasks.



(**a**) NWPU-RESISC45 (**b**) RSD46-WHU

(**c**) UC Merced (**d**) WHU-RS19

Figure 5. Influences of the pre-trained feature extractor.

4.4.2. Performance Analysis of Different Classifiers

To solve the problem of linear inseparability, we used the kernel space classifier instead of the traditional linear classifier to map the sample features into a high-dimensional kernel space. In this section, we perform ablation experiments to investigate the effects of different kernel space classifiers on classification performance. We use three different kernels—the linear kernel, the poly kernel, and the RBF kernel—and compare the results of the logistic regression (LR) classifier and support vector machine (SVM) classifier. The experimental results of four remote sensing datasets are listed in Tables 6–9.

It can be seen that the performance of the RBF and poly-kernel space classifier are better than that of the linear kernel. Specifically, in the case of 5-way 5-shot, the performances of the poly-kernel space classifier on NWPU-RESISC45 and WHU-RS19 datasets are the best. Still, for the RSD46-WHU and UC Merced datasets, the performance of the RBF kernel space classifier is the best. In the case of 5-way 1-shot, the RBF kernel space classifier achieved satisfactory performance in all four datasets.

Moreover, the experimental results can prove that the classification performance of the kernel space classifier is better than that of the traditional classifier (e.g., LR or SVM). To be specific, for the four remote sensing datasets, the performance is improved by 0.35%∼2.71% using the kernel space classifier in the case of 5-way 5-shot, and 1.75%∼6.47% in the case of 5-way 1-shot.

Table 6. Comparison results of the different methods in the NWPU-RESISC dataset.

Method	Backbone	5-Way 1-Shot	5-Way 5-Shot
LR	ResNet-12	69.68 ± 0.78	87.18 ± 0.44
SVM	ResNet-12	67.56 ± 0.80	86.20 ± 0.45
Linear	ResNet-12	73.82 ± 0.77	87.95 ± 0.43
Poly	ResNet-12	73.84 ± 0.76	88.36 ± 0.43
RBF	ResNet-12	74.03 ± 0.76	88.32 ± 0.43

Table 7. Comparison results of different methods in the RSD46-WHU dataset.

Method	Backbone	5-Way 1-Shot	5-Way 5-Shot
LR	ResNet-12	70.95 ± 0.90	86.98 ± 0.53
SVM	ResNet-12	68.28 ± 0.93	85.92 ± 0.55
Linear	ResNet-12	73.92 ± 0.88	87.60 ± 0.51
Poly	ResNet-12	73.38 ± 0.89	88.07 ± 0.50
RBF	ResNet-12	74.10 ± 0.88	88.19 ± 0.52

Table 8. Comparison results of different methods in the UC Merced dataset.

Method	Backbone	5-Way 1-Shot	5-Way 5-Shot
LR	ResNet-12	63.69 ± 0.77	80.58 ± 0.50
SVM	ResNet-12	64.00 ± 0.76	80.07 ± 0.51
Linear	ResNet-12	64.70 ± 0.73	80.78 ± 0.47
Poly	ResNet-12	64.88 ± 0.73	82.38 ± 0.46
RBF	ResNet-12	65.44 ± 0.72	82.78 ± 0.45

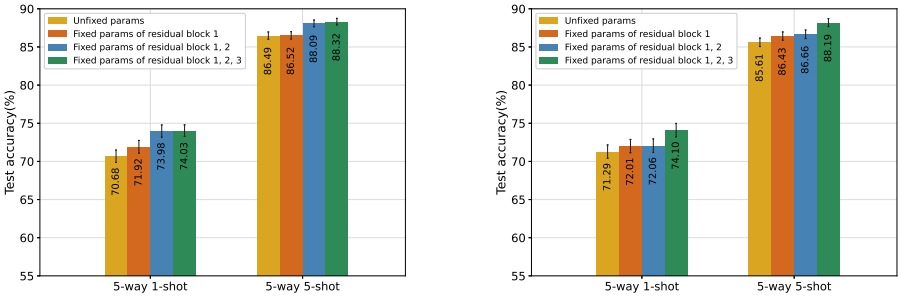
Table 9. Comparison results of different methods in the WHU-RS19 dataset.

Method	Backbone	5-Way 1-Shot	5-Way 5-Shot
LR	ResNet-12	80.74 ± 0.60	93.27 ± 0.30
SVM	ResNet-12	79.56 ± 0.62	92.88 ± 0.30
Linear	ResNet-12	82.43 ± 0.58	92.18 ± 0.24
Poly	ResNet-12	82.51 ± 0.59	93.62 ± 0.24
RBF	ResNet-12	82.57 ± 0.58	93.33 ± 0.23

4.4.3. Influences of Different Fine-Tuning Strategies

To make the pre-trained feature extractor suitable for the remote sensing scene classification task, we used the corresponding remote sensing dataset to fine-tune the parameters of the pre-trained feature extractor. We believe that the pre-trained model extracts prior knowledge from the training set, and this high-order model can be used for the feature extraction of various tasks. In the process of fine-tuning the model, The last layers are altered to encode specific features of remote sensing data, while the earlier layers are kept since they encode more general features.

The ablation experiments determine the number of residual blocks with fixed parameters. Figure 6 shows the comparison of the results in four cases: (1) fine-tuning the parameters of four residual blocks; (2) fixing the parameters of the first residual block and fine-tuning the parameters of the remaining residual blocks; (3) fixing the parameters of the first two residual blocks and fine-tuning the parameters of the remaining residual blocks; (4) fixing the parameters of the first three residual blocks and fine-tuning the parameters of the last residual block only. The experimental results in the NWPU-RESISC45 and RSD46-WHU datasets show that the feature extractor achieves the best performance by only fine-tuning the last residual block.



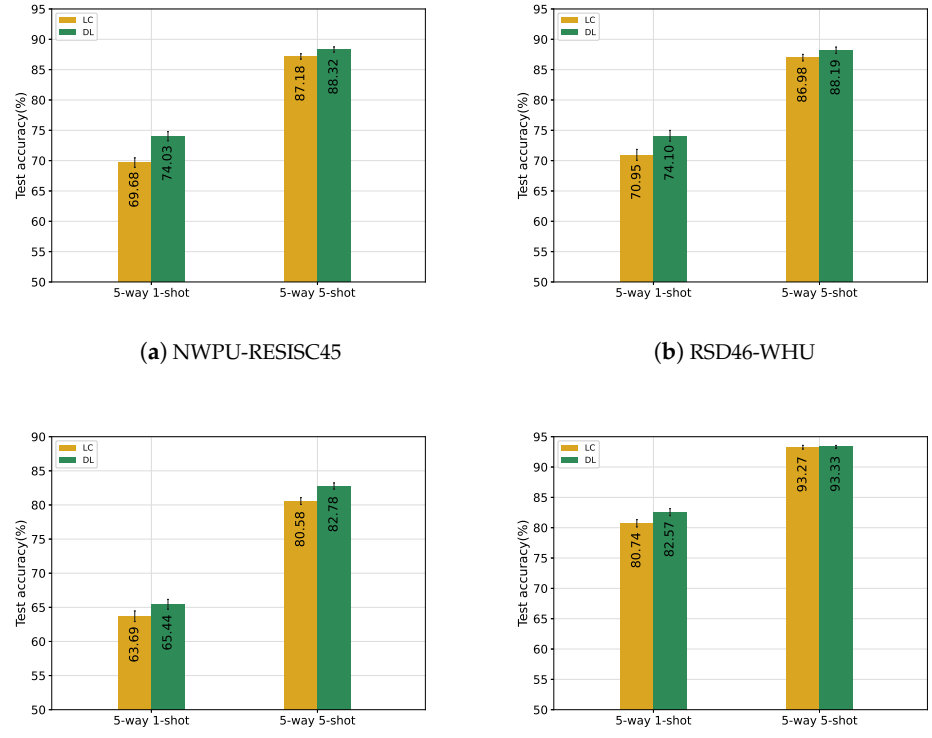
(**a**) NWPU-RESISC45 (**b**) RSD46-WHU

Figure 6. Influence of different fine-tuning strategies.

4.4.4. Influence of the Objective Function Reconstruction Error

In Formula (1), the objective function includes two parts: reconstruction error and classification error. We performed ablation experiments to explore the benefits of the reconstruction part of the classification task. When the reconstruction error part of the objective function is removed, and only the classification error is used to complete the classification task, it is equivalent to a linear regression classifier (LC). The experimental results are shown in Figure 7.

For the four remote sensing datasets, the results of our proposed method LC reach 69.68%, 70.95%, 63.69%, and 80.72% in the 1-shot case, and 87.18%, 86.98%, 80.58%, and 93.27% in the 5-shot case, respectively, achieving worse performances than the LC. The experimental results show that the classification performance will be reduced to different degrees when only the classification error is used, and the reconstruction error is helpful in improving the classification performance further.



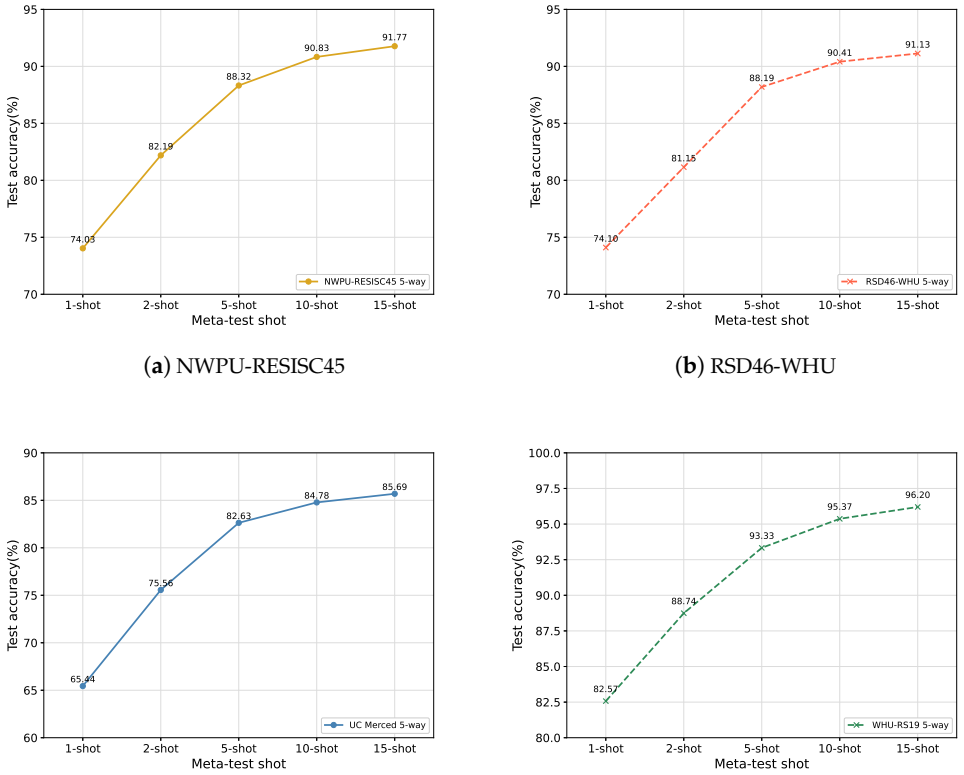
(**a**) NWPU-RESISC45 (**b**) RSD46-WHU

(**c**) UC Merced (**d**) WHU-RS19

Figure 7. The influence of objective function reconstruction error.

4.4.5. Influence of Meta-Test Shot

To explore the influences of different shots on the performance, we fix 'the way' as 5 and conduct experiments on four datasets in the cases of 1-shot, 2-shot, 5-shot, 10-shot, and 15-shot, respectively. The results are shown in Figure 8. It can be seen that with the increase of the shot, the performance of our method gradually improves, but the speed of improvement gradually slows down, and the optimal performance is achieved at the 15-shot.



(**a**) NWPU-RESISC45 (**b**) RSD46-WHU

(**c**) UC Merced (**d**) WHU-RS19

Figure 8. Comparison of different shots for the meta-test on four datasets: (**a**) NWPU-RESISC45 (**b**) RSD46-WHU (**c**) UC Merced (**d**) WHU-RS19

5. Discussion

The proposed method achieves competitive performance through experiments on multiple benchmark datasets compared with the state-of-the-art few-shot remote sensing scene classification methods. Although the proposed method improves the few-shot remote sensing scene classification performance, it has the following limitation. The proposed method has three parameters that need to be adjusted manually in the meta-test stage, and different parameters correspond to various performances, limiting the availability of the method. In the future, we plan to adopt the self-training method to expand and improve the proposed method, extending the generalized few-shot remote sensing scene classification task to the transductive few-shot remote sensing scene classification task. Moreover, we will explore nonlinear base learners for future work, such as kernel methods.

6. Conclusions

In this paper, we propose dictionary learning for the few-shot remote sensing scene classification method, which adopts the kernel space classifier to map the features to a high-dimensional space and embed the label information into the dictionary learning to improve the discrimination of features for classification. In addition, we suggest using the pre-trained feature extractor with fine-tuning to make the feature extractor suitable for remote

sensing data. Extensive experiments on four popular remote sensing scene classification datasets demonstrate the effectiveness of our proposed dictionary learning method.

Author Contributions: Conceptualization, Y.M., J.M. and B.L.; methodology, Y.M., L.S., P.R., J.M. and B.L.; validation, Y.M., L.S., P.R., J.M. and B.L.; formal analysis, Y.M., L.S., P.R., J.M. and H.Z.; investigation, Y.M., L.S., P.R., B.L. and H.Z.; writing—original draft preparation, Y.M. and J.M.; writing—review and editing, B.L.; visualization, Y.M. and H.Z.; supervision, Y.M., L.S., P.R. and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The paper was supported by the National Natural Science Foundation of China: no. 42006164 and the National Natural Science Foundation of China: no. U2006207.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author.

Acknowledgments: We would like to express our gratitude to the editor and reviewers for their valuable comments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FSRSSC	few-shot remote sensing scene classification
RSSC	remote sensing scene classification
FSL	few-shot learning
HOG	histograms of oriented gradients
SIFT	scale-invariant feature transform
LBP	local binary pattern
BovW	bag-of-visual-words
LR	logistic regression
SVM	support vector machine
DL	dictionary learning

References

1. Johnson, B.A.; Jozdani, S.E. Local Climate Zone (LCZ) Map Accuracy Assessments Should Account for Land Cover Physical Characteristics that Affect the Local Thermal Environment. *Remote Sens.* **2019**, *11*, 2420. [CrossRef]
2. Pham, H.M.; Yamaguchi, Y.; Bui, T.Q. A case study on the relation between city planning and urban growth using remote sensing and spatial metrics. *Landsc. Urban Plan.* **2011**, *100*, 223–230. [CrossRef]
3. Zhu, Q.; Zhong, Y.; Zhao, B.; Xia, G.-S.; Zhang, L. Bag-of-Visual-Words Scene Classifier with Local and Global Features for High Spatial Resolution Remote Sensing Imagery. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 747–751. [CrossRef]
4. Manfreda, S.; McCabe, M.F.; Miller, P.E.; Lucas, R.; Pajuelo Madrigal, V.; Mallinis, G.; Ben Dor, E.; Helman, D.; Estes, L.; Ciraolo, G.; et al. On the Use of Unmanned Aerial Systems for Environmental Monitoring. *Remote Sens.* **2018**, *10*, 641. [CrossRef]
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
6. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N.; Zhan, Y. Semi-supervised locality preserving dense graph neural network with ARMA filters and context-aware learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5511812. [CrossRef]
7. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Multiscale graph sample and aggregate network with context-aware learning for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4561–4572. [CrossRef]
8. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Graph sample and aggregate-attention network for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 5504205. [CrossRef]
9. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, W.; He, F.; Cai, Y.; Cai, W. Deep hybrid: Multi-Graph neural network collaboration for hyperspectral image classification. *Def. Technol.* 2022, *in press*. [CrossRef]
10. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. *arXiv* **2017**, arXiv:1703.05175.

11. Shao, S.; Xing, L.; Xu, R.; Liu, W. MDFM: Multi-Decision Fusing Model for Few-Shot Learning. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 5151–5162. [CrossRef]
12. Wang, Y.; Xu, C.; Liu, C.; Zhang, L.; Fu, Y. Instance Credibility Inference for Few-Shot Learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12833–12842.
13. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
14. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
15. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
16. Yang, Y.; Newsam, S. Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. Association for Computing Machinery, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
17. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Cai, W.; Yu, C.; Yang, N.; Cai, W. Multi-feature Fusion: Graph Neural Network and CNN Combining for Hyperspectral Image Classification. *Neurocomputing* **2022**, *501*, 246–257. [CrossRef]
18. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Li, W.; Cai, W.; Zhan, Y. AF2GNN: Graph convolution with adaptive filters and aggregator fusion for hyperspectral image classification. *Inf. Sci.* **2022**, *602*, 201–219. [CrossRef]
19. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, Y.; Li, S.; Deng, B.; Cai, W. Self-supervised locality preserving low-pass graph convolutional embedding for large-scale hyperspectral image clustering. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5536016. [CrossRef]
20. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, W.; Yang, N.; Hu, H.; Huang, X.; Cao, Y.; Cai, W. Unsupervised self-correlated learning smoothy enhanced locality preserving graph convolution embedding clustering for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5536716. [CrossRef]
21. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the ICML International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
22. Nichol, A.; Achiam, J.; Schulman, J. On First-Order Meta-Learning Algorithms. *arXiv* **2018**, arXiv:1803.02999.
23. 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.
24. Alajaji, D.A.; Alhichri, H. Few Shot Scene Classification in Remote Sensing using Meta-Agnostic Machine. In Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 4–5 March 2020; pp. 77–80.
25. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 2.
26. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *NeurIPS* **2016**, *29*, 3630–3638.
27. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.; Hospedales, T.M. Learning to compare: Relation network for few-shot learning. In Proceedings of the CVPR IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1199–1208.
28. 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.* **2021**, *59*, 7844–7853. [CrossRef]
29. Alajaji, D.; Alhichri, H.S.; Ammour, N.; Alajlan, N. Few-shot learning for remote sensing scene classification. In Proceedings of the M2GARSS, 2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), Tunis, Tunisia, 9–11 March 2020; IEEE: New York, NY, USA, 2020; pp. 81–84.
30. Zhang, P.; Bai, Y.; Wang, D.; Bai, B.; Li, Y. Few-Shot Classification of Aerial Scene Images via Meta-Learning. *Remote Sens.* **2021**, *13*, 108. [CrossRef]
31. Dvornik, N.; Schmid, C.; Mairal, J. Diversity with Cooperation: Ensemble Methods for Few-Shot Classification. *arXiv* **2019**, arXiv:1903.11341.
32. Yue, Z.; Zhang, H.; Sun, Q.; Hua, X.S. Interventional Few-Shot Learning. *arXiv* **2020**, arXiv:2009.13000.
33. Shao, S.; Xing, L.; Wang, Y.; Xu, R.; Zhao, C.; Wang, Y.; Liu, B. MHFC: Multi-Head Feature Collaboration for Few-Shot Learning. In Proceedings of the ACM MM, 29th ACM International Conference on Multimedia, Virtual Event, China, 20–24 October 2021; pp. 4193–4201.
34. Lee, K.; Maji, S.; Ravichandran, A.; Soatto, S. Meta-learning with differentiable convex optimization. In Proceedings of the CVPR 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10657–10665.
35. Ren, M.; Ravi, S.; Triantafillou, E.; Snell, J.; Swersky, K.; Tenenbaum, J.B.; Larochelle, H.; Zemel, R.S. Meta-Learning for Semi-Supervised Few-Shot Classification. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
36. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]

37. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]
38. Xiao, Z.; Long, Y.; Li, D.; Wei, C.; Tang, G.; Liu, J. High-Resolution Remote Sensing Image Retrieval Based on CNNs from a Dimensional Perspective. *Remote Sens.* **2017**, *9*, 725. [CrossRef]
39. Xia, G.S.; Yang, W.; Delon, J.; Gousseau, Y.; Sun, H.; Maitre, H. Structural high-resolution satellite image indexing. In Proceedings of the ISPRS TC VII Symposium—100 Years ISPRS, Vienna, Austria, 5–7 July 2010.
40. Dai, D.; Yang, W. Satellite Image Classification via Two-Layer Sparse Coding With Biased Image Representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *8*, 173–176. [CrossRef]
41. Zhai, M.; Liu, H.; Sun, F. Lifelong Learning for Scene Recognition in Remote Sensing Images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1472–1476. [CrossRef]
42. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv* **2017**, arXiv:1707.09835.
43. Oreshkin, B.N.; Rodriguez, P.; Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the NeurIPS, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, QC, Canada, 2–8 December 2018; pp. 719–729.
44. 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.* **2021**, *13*, 4200. [CrossRef]
45. 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. [CrossRef]
46. Simon, C.; Koniusz, P.; Nock, R.; Harandi, M. Adaptive subspaces for few-shot learning. In Proceedings of the CVPR, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4136–4145.
47. Liu, Y.; Lee, J.; Park, M.; Kim, S.; Yang, E.; Hwang, S.J.; Yang, J. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv* **2018**, arXiv:1805.10002
48. Zhang, P.; Li, Y.; Wang, D.; Wang, J. RS-SSKD: Self-Supervision Equipped with Knowledge Distillation for Few-Shot Remote Sensing Scene Classification. *Sensors* **2021**, *21*, 1566. [CrossRef]
49. Ye, H.J.; Hu, H.; Zhan, D.C.; Sha, F. Few-shot learning via embedding adaptation with set-to-set functions. In Proceedings of the CVPR, 2020 Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 8808–8817.
50. Cui, Z.; Yang, W.; Chen, L.; Li, H. MKN: Metakernel networks for few shot remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4705611. [CrossRef]
51. Yuan, Z.; Huang, W. Multi-attention DeepEMD for Few-Shot Learning in Remote Sensing. In Proceedings of the ITAIC, 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; IEEE: New York, NY, USA, 2020; Volume 9, pp. 1097–1102.
52. Zhang, C.; Cai, Y.; Lin, G.; Shen, C. DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover's Distance and Structured Classifiers. In Proceedings of the CVPR, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020 2020; pp. 12203–12213.
53. Li, H.; Cui, Z.; Zhu, Z.; Chen, L.; Zhu, J.; Huang, H.; Tao, C. RS-MetaNet: Deep meta metric learning for few-shot remote sensing scene classification. *arXiv* **2020**, arXiv:2009.13364.