Article

Task-Adaptive Embedding Learning with Dynamic Kernel Fusion for Few-Shot Remote Sensing Scene Classification

Pei Zhang [1], Guoliang Fan[2], Chanyue Wu [1], Dong Wang [1], and Ying Li [1,*]

[1] School of Computer Science, National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, Shaanxi Provincial Key Laboratory of Speech & Image Information Processing, Northwestern Polytechnical University, Xi'an 710129, China; cszhangpei@mail.nwpu.edu.cn (P.Z.); wuchanyuec@163.com (C.W.); dongwang@mail.nwpu.edu.cn (D.W.); lybyp@nwpu.edu.cn (Y.L.)

[2] School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, Oklahoma, United States, 74078; guoliang.fan@okstate.edu (G.F.)

* Correspondence: lybyp@nwpu.edu.cn (Y.L.); Tel.: +86-029-8843-1532

Abstract: The central goal of few-shot scene classification is to learn a model that can generalize well to a novel scene category (UNSEEN) from only one or a few labeled examples. Recent works in the remote sensing (RS) community tackle this challenge by developing algorithms in a meta-learning manner. However, most prior approaches have either focused on rapidly optimizing a meta-learner or aimed at finding good similarity metrics while overlooking the embedding power. Here we propose a novel Task-Adaptive Embedding Learning (TAEL) framework that complements the existing methods by giving full play to feature embedding's dual roles in few-shot scene classification - representing images and constructing classifiers in the embedding space. First, we design a lightweight network that enriches the diversity and expressive capacity of embeddings by dynamically fusing information from multiple kernels. Second, we present a task-adaptive strategy that helps to generate more discriminative representations by transforming the universal embeddings into task-specific embeddings via a self-attention mechanism. We evaluate our model in the standard few-shot learning setting on two challenging datasets: NWPU-RESISC4 and RSD46-WHU. Experimental results demonstrate that, on all tasks, our method achieves state-of-the-art performance by a significant margin.

Keywords: remote-sensing classification; scene classification; few-shot learning; meta-learning; vision transformers; multi-scale feature fusion

1. Introduction

Scene classification plays an essential role in the semantic understanding of remote sensing (RS) images by classifying each image into different categories according to its contents [1]. It provides valuable support to applications ranging from land use and land cover (LULC) determination [2,3], environmental monitoring [4], urban planning [5,6] and deforestation mapping [7].

In the past few years, deep learning-based approaches [8–12] have achieved human-level performance on certain RS scene classification benchmarks [1,13–15]. Despite the remarkable achievements, these excellent methods are data-hungry in order to learn massive parameters and often fail when encountering the natural conditions that humans face in the real world - data is not always enough. For instance, consider training a traditional classifier to identify a novel category that has never existed in the current RS scene datasets, e.g., bicycle-sharing parking lot, a new scene that has recently emerged in China. One would have to first collect hundreds or thousands of relevant RS images taken from the air and space. The high cost of collecting and annotating hinders many downstream applications where data is inherently rare or expensive. Moreover, a trained deep learning model usually struggles when asked to solve a new task unless it re-executes the training process with high computational cost. In contrast, humans can learn new concepts quickly from just one, or a handful examples by drawing upon previous knowledge and experience [**?**]. These issues motivated research on few-shot learning (FSL) [**? ? ?**] - a learning paradigm

Citation: 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. *Preprints* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

 doi:10.20944/preprints202108.0389.v1

that emulates human learning - the ability to learn and adapt to new environments rapidly. Specifically, the contemporary FSL setting [**? ? ? ?**] is designed to mimic a low-data scenario. Focusing on few-shot classification tasks, we are dealing with two sets of categories - base set (SEEN) and novel set (UNSEEN) - that disjoint in the label space. A successful FSL learner needs to exploit transferable knowledge in the base set, which has sufficient labeled data, and leverage it to build a classifier that generalizes well on UNSEEN categories when provided with extremely few labeled instances per category, e.g., 1 or 5 images. Recent research generally addresses the FSL problem by following the idea of meta-learning, i.e., broadening the learner's scope to batches of related tasks/episodes rather than batches of data points, and gains experience across the tasks. This episodic training scheme is also referred to as *learning-to-learn* by leveraging the experience to improve the future learning performance.

The recent success of few-shot learning has captured attention in the remote sensing community. Rußwurm et al. [16] evaluate a well-known meta-learning algorithm, model-agnostic meta-learning (MAML) [**?**], for land cover few-shot classification problems. They observe that MAML outperforms the traditional transfer learning methods. The work [17] adopts deep few-shot learning to handle the small sample size problem in hyperspectral image classification. Most previous RS scene few-shot classification methods [18**?** –20] fall under the umbrella of metric learning and are built upon Prototypical Networks (ProtoNet) [**?**]. RS-MetaNet [20] improves ProtoNet with a new balance loss that combines the maximum generalization loss and the cross-entropy loss. Zhang et al. [18] present a meta-learning framework based on ProtoNet and use cosine distance with a learnable scale parameter to achieve better performance. Later on, DLA-MatchNet [21] couples the attention technique and Relation Network [**?**], where the former aims to exploit discriminative regions while the latter learns the similarity scores between the images by an adaptive matcher. RS-SSKD [**?**] proposes a self-supervision strategy to drive the network digging the most discriminative category-specific region and boost the performance by a round of self-knowledge distillation. While these methods have achieved significant progress in RS few-shot classification, we observe that these approaches do suffer from two distinct limitations.

One missing piece of the puzzle is that these metric-based algorithms mainly focus on identifying a suitable similarity measure or construct a combined loss function to drive the parameter updates while overlooking the importance of the embedding network. DLA-MatchNet [21] introduces an attention mechanism in the feature learning stage to capture attention features from channels and spatial dimensions. RS-SSKD [**?**] weaves self-supervision into a two branches network to dig the base-set data fully by refining the pre-training embedding. Both methods aim at learning the most relevant regions to get better embeddings. On the other hand, we pay attention to the inherent characteristics of remote sensing data. For example, as the RS scene images are taken from a top view, the ground objects vary from small sizes such as airplanes to large regions like a forest or meadow. Moreover, under a spatial resolution range from about 30 to 0.2m per pixel (e.g., the NWPU-RESISC45 dataset [15]), irrelevant objects inevitably exist in the RS scene images (see Figure 1). These issues may drive the embeddings from the same category far apart in a given metric space. If we have sufficient training samples, this problem can be greatly alleviated by a deeper neural network. However, we are dealing with a low data regime of the FSL setting, where the embedding network is either too shallow to leverage the model's expressive capacity or too deep and results in overfitting [**?**]. That is the reason why Conv-4 [**? ? ? ?**], Resnet-12, and Resnet-18 [**?**] are the most popular embedding networks in the FSL world. The other concern is that the existing models generally project all instances from various tasks into a single common embedding space indiscriminately [18**?** –21]. Such strategy implies that the discovered knowledge, i.e., embedded visual features learned on the SEEN categories, are equally useful for any downstream target classification tasks derived from UNSEEN categories. We argue that the issue of which features are the most discriminative to a specific target task has not received considerable

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

Figure 1. Examples of the inherent characteristics of remote sensing scene images, i.e., the ground objects vary in size and irrelevant objects exist.

attention. For instance, consider that we have two separate classification tasks: "freeway" vs. "forest" and "freeway" vs. "bridge". It is intuitive that these two tasks use a diverse set of most discriminative features. Therefore, the ideal embedding model would first need to extract discriminative features for either task simultaneously, which is challenging. Since the current model does not know what exactly the "downstream" target tasks are, it may unexpectedly emphasize unimportant features for later use. Further, even if two sets of discriminative features are extracted, they do not certainly head to the best performance for a specific target task. For example, the most useful feature to distinguish "freeway" vs. "forest" may be irrelevant to the task of distinguishing "freeway" vs. "bridge". Naturally, we expect the embedding spaces are separated, where each of which is customized to the target task so that the extracted visual features are the most discriminative. Figure 2 schematically illustrates the difference between task-agnostic and task-adaptive embeddings.

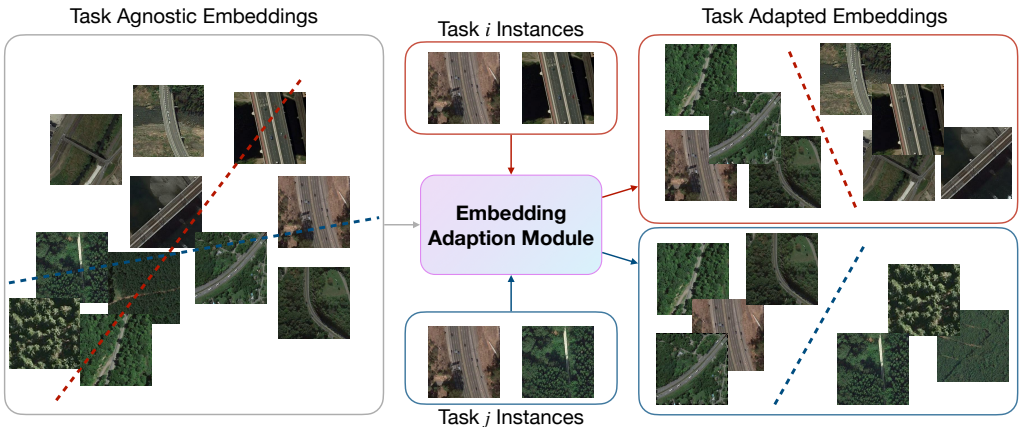

Figure 2. An illustration of the difference between task-agnostic and task-adaptive embeddings.

To sum up, we suggest that the embedding module is crucial due to its dual roles - representing inputs and constructing classifiers in the embedding space. Several recent studies [**? ? ?**] have supported this assumption with a series of experiments and verified that better embeddings head to better few-shot learning performance. The question is, how to get a good embedding? We answer this question by solving two challenges: 1) design a lightweight embedding network that tackles the problems posed by the inherent characteristics of RS scene images; 2) construct an embedding adaptation module that tailors the common embeddings into adaptive embeddings according to a specific target task. See Section 3.3 and 3.4 for details.

Our main contributions in this paper are summarized as follows:

- We develop an efficacious meta-learning scheme that utilizes two insights to improve few-shot classification performance: a lightweight embedding network that

captures multi-scale information and a task-adaptive strategy that further refines the embeddings.
- We present a new embedding network - Dynamic Kernel Fusion Network (DKF-Net) - that dynamically fuses feature representations from multiple kernels while preserving comparably lightweight customization for few-shot learning.
- We propose a novel embedding adaptation module that transforms the universal embeddings obtained from the base set into task-adaptive embeddings via a self-attention architecture. This is achieved by a set-to-set function that contextualizes the instances over a set to ensure that each has strong co-adaptation.
- The experimental results on two remote sensing scene datasets demonstrate that our framework surpasses other state-of-the-art approaches. Furthermore, we offer extensive ablation studies to show how the choices in our training scheme impact the few-shot classification performance.

The rest of this paper is organized as follows. We review the related work in Section 2. The problem setting and the proposed framework are formally described in Section 3. We report the experimental results in Section 4 and discuss with ablation studies in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Current few-shot learning has been primarily addressed in the meta-learning manner, where a model is optimized through batches of training episodes/tasks rather than batches of data points, which is referred to as episodic training [**?**]. We can roughly divide the existing works on FSL into two groups. (1) Optimization-based methods search for more transferable representations with sensitive parameters that could rapidly adapt to new tasks in the meta-test stage within a few gradient descent steps. MAML [**?**], Reptile [**?**], LEO [**?**], and MetaOptNet [**?**] are the most representative approaches in this family. (2) Metric-based methods mainly learn to represent input data in an appropriate embedding space, where a query sample is easy to classify with a distance-based prediction rule. One can measure the distance in the embedding space by simple distance functions such as cosine similarity (e.g., Matching Network [**?**]) or Euclidean distance (e.g., Protypical Networks [**?**]), or learn parameterized metrics via an auxiliary network (e.g., Relation Network [**?**]). Later, in DSN-MR [**?**], all samples and metrics are operated in affine subspaces. SAML [**?**] suggests the global embeddings are not discriminative enough as dominant objects may locate anywhere on images. The authors tackle this problem by a "collect-and-select" strategy that aligns the relevant local regions between query and support images in a relation matrix. Given the local feature sets generated by two images, DeepEMD [**?**] shoots the same problem by employing the Earth Mover's Distance [**?**] to capture their structural similarity. Our work falls in the second group but differs from them in two folds.

First, like SAML and DeepEMD, Tian et al. [**?**] also suggest that the core of improving FSL lies in learning more discriminative embedding. In response, contemporary approaches address this challenge either refining the pre-training strategy to exploit the base-set data fully [**?**], leveraging self-supervision to feed auxiliary versions of original images into the embedding network [**?**], or applying self-distillation to achieve an additional boost [**?**]. While these approaches effectively make the embedding more representative, they tend to concentrate too much on designing a complex loss function [**?** **?**] or building networks to capture relevant local features at the cost of computing resources and time [**?** **?**]. On the contrary, our solution offers a lightweight embedding network that generates more discriminative representations while imposing fewer parameters than the most popular backbone, i.e., ResNet-12 [**?** **?**], in few-shot learning. A fundamental property of neurons present in the visual cortex is changing their receptive fields (RF) in response to the stimulus [**?**]. This mechanism of adaptively adjusting receptive fields can be incorporated in neural networks by multi-scale feature aggregation and selection, which would benefit constructing a desirable RS scene few-shot classification algorithm - considering that the

 doi:10.20944/preprints202108.0389.v1

ground objects vary largely in size. Inspired by Selective Kernel (SK) Networks [**?**], we introduce a nonlinear procedure for fusing features from multiple kernels in the same layer by a self-attention mechanism. We incorporate two-branches SK convolution into our embedding network and name it Dynamic Kernel Fusion Network (DKF-Net).

Second, the abovementioned methods assume all samples are embedded into a task-agnostic space, hoping the embeddings could sufficiently represent the support data such that the similarities predicted from simple non-parametric classifiers will generalize well to new tasks. We suggest that ideal embedding spaces for few-shot learning should be separated, where each of them is customized to the target task adaptively so that the extracted visual features are discriminative. Some recent works also pay attention to this assumption. TADAM [**?**] proposes to learn a task-dependent metric space by constructing a conditional learner on the task-level set and optimizing with an auxiliary task co-training procedure. TapNet [**?**] constructs a projection space for each episode/task and introduces additional reference vectors, in which the class prototypes and the reference vectors are closely aligned. Unfortunately, the task-dependent conditioning mechanism in TADAM requires learning of extra fully connected networks while the projection space in TapNet is solved through the singular value decomposition (SVD) step; both strategies significantly increase training time. Taking inspiration from Transformer [**?**], we propose an embedding adaption module based on a self-attention mechanism that transforms *task-agnostic* embeddings into *task-adaptive* embeddings, see Section 3.4 and Figure 5.

3. Methodology

We now present our approach for the few-shot classification of RS scenes, starting with preliminaries. Then, we present our few-shot learning workflow in Section 3.2, wherein the overall framework is depicted in Figure 3. The proposed Dynamic Kernel Fusion Network (DKF-Net) is described in Section 3.3, the embedding backbone in the whole flow of our work. At last, we elaborate on the embedding adaption module and discuss how it helps few-shot learning in Section 3.4.

3.1. Preliminaries

Problem setting. In traditional classification setting, we are given a dataset $D = \{D_{train}, D_{test}\}$ with C_{total} categories. $D_{train} = \{(x_i, y_i)\}_{i=1}^{N}$ terms as the training set, where $y_i \in \{1, ..., C_{total}\}$ and (x_i, y_i) is the input image and corresponding label pairs. A predictive model is learned on D_{train} at training time, and generalization is then evaluated on D_{test}, i.e., the test set. In few-shot learning (FSL), however, we are dealing with a dataset $\mathcal{D}$, divided into three parts with respect to categories: $\mathcal{D}_{base}$, $\mathcal{D}_{val}$, and $\mathcal{D}_{novel}$, i.e., training set, validation set, and test set. The category spaces in the three sets are disjoint with each other. The goal of FSL is to learn a general-purpose model on $\mathcal{D}_{base}$(SEEN) that can generalize well to UNSEEN categories in $\mathcal{D}_{novel}$ with one or few training instances per category. In addition, $\mathcal{D}_{val}$ is held out to select the best model.

Episodic training. To mimic the low-data scenario during testing, most of the FSL methods [**? ? ? ? ? ?**] proceed in a meta-learning fashion. The intuition behind meta-learning is improving the performance of a model by extracting transferable knowledge from a collection of sampled mini-batches called *episodes*, a.k.a, *tasks*, and minimizing the generalization error over a task distribution. Formally, a set of M tasks is denoted as $\mathcal{T} = \{(\mathcal{S}_i, \mathcal{Q}_i)\}_{i=1}^{M}$, sampled from a task distribution $p(\mathcal{T})$. Each task $\mathcal{T}_i$ can be considered a compact dataset containing both training and test data, referred to as support set $\mathcal{S}_i$ and query set $\mathcal{Q}_i$.

3.2. Overall framework

The outline of our method to RS scene few-shot classification is: (1) We employ a pre-training stage to learn an embedding model $f_\phi(x)$ on the base set $\mathcal{D}_{base}$; (2) in the meta-learning stage, we optimize the embedding model with the nearest centroid classifier, in an episodic meta-learning paradigm; (3) at inference time, i.e., the meta-test stage, the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

model is fixed, we sample tasks from the novel set $\mathcal{D}_{novel}$ for evaluation and report the mean accuracy. The overview of our method is depicted in Figure 3. All the stages of our model are built upon the proposed DFK-Net backbone (see Section 3.3 and Figure 4). The details of these stages are as follows.

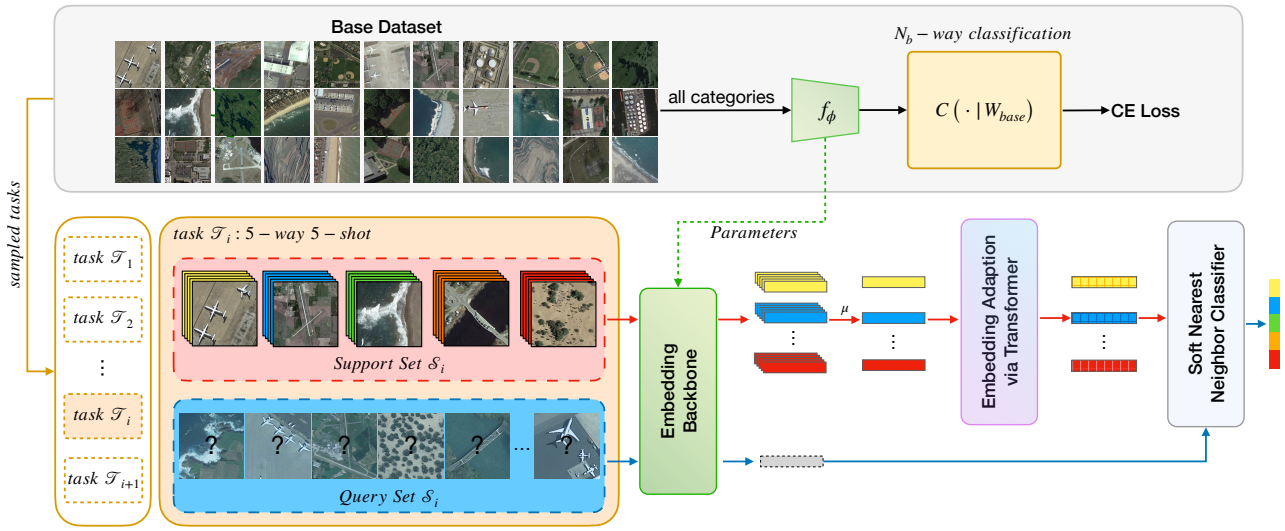

Figure 3. Overall framework of the proposed method.

Pre-training stage. We train a base classifier on $\mathcal{D}_{base}$ to learn a general feature embedding for the downstream meta-leaner, which is helpful to yield robust few-shot classification. The predictive model $\hat{y} = f_\phi(x)$, parameterized by ϕ, is trained to classify N_b base categories (e.g., 25 categories in the NWPU-RESISC45 Dataset) in $\mathcal{D}_{\text{base}}$ with the standard cross-entropy (CE) loss, by solving:

$$\phi = \underset{\phi}{\operatorname{argmin}} \mathcal{L}_{ce}(\mathcal{D}_{\text{base}}; \phi). \tag{1}$$

The performance of the pre-trained model is evaluated after each epoch, based on its 1-shot classification accuracy on the validation set $\mathcal{D}_{val}$. Specifically, assuming that there are N_v categories in $\mathcal{D}_{val}$, we randomly sample 200 1-shot N_v-way tasks from $\mathcal{D}_{val}$ to assess the classification performance of the pre-trained model and select the best one. The weights of the penultimate layer from the best pre-trained model are utilized to initialize the embedding backbone and are optimized in the next meta-learning stage.

Meta-learning stage. In most few-shot learning setups, a model is often evaluated in N-way K-shot tasks, K is usually very small, e.g., $K = 1$ or $K = 5$ is the most common setting. Following prior work [**? ? ?**], an N-way K-shot task $\mathcal{T}_i$ is constructed by randomly sampling N categories, and K labeled instances per category as the support set $\mathcal{S}_i = \{(x_n, y_n)\}_{n=1}^{N\times K}$, where (x_n, y_n) is an image-label pair, and $y_n \in \{1, \ldots, N\}$. We take a fraction of the remaining instances from the same N categories to form the query set $\mathcal{Q}_i = \{(x_n, y_n)\}_{n=1}^{N\times Q}$, and the end goal becomes the classification of the $N \times Q$ unlabelled instances into N categories. Note that, $\mathcal{S}_i$ and $\mathcal{Q}_i$ are disjoint, i.e., $\mathcal{S}_i \cap \mathcal{Q}_i = \varnothing$ while sharing the same label space. Since the pre-trained model is trained only on the base set, it often falls into the over-fitting dilemma or is updated very little when facing the novel categories with a meager amount of support instances. Some recent approaches handle this problem by fixing the pre-trained model and finetune it on the novel set. We adopt an opposite strategy by using a meta-learning paradigm built upon ProtoNet [**?**] to optimize the pre-trained model f_ϕ, parameterized by ϕ, directly without introducing any extra parameters.

During the meta-learning stage, we sample a collection of N-way K-shot tasks $\{\mathcal{T}_i = (\mathcal{S}_i, \mathcal{Q}_i)\}_{i=}^{I}$ from $\mathcal{D}_{base}$ to form the meta-training set $\mathcal{T}^{\text{train}}$. Likewise, we obtain the meta-validation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

set $\mathcal{T}^{\text{val}}$ and the meta-test set $\mathcal{T}^{\text{test}}$ from $\mathcal{D}_{val}$ and $\mathcal{D}_{novel}$ in the same way. Given the meta-training set $\mathcal{T}^{\text{train}}$, the meta-learning procedure minimizes the generalization error across tasks. Thus, the learning objective can be loosely defined as:

$$\phi = \underset{\phi}{\text{argmin}} \mathbb{E}_{\mathcal{T}^{\text{train}}} \left[\mathcal{L}_{\text{meta}} \left(Q; \phi \right) \right]. \tag{2}$$

For each N-way K-shot task $\mathcal{T}_i = (\mathcal{S}_i, \mathcal{Q}_i)$, there are K images belong to category c in the support set, where $c \in \{1, \ldots, N\}$. We define the mean feature of these K images as the *prototype* $\mathbf{p}_c$, i.e., the category center, corresponding to category c:

$$\mathbf{p}_c = \frac{1}{|K|} \sum_{(x_k, y_k) \in \mathcal{S}_i, y_k = c} f_\phi(\mathbf{x}_k), \tag{3}$$

where f_ϕ is an embedding function with learnable parameters ϕ, mapping the input x_k into the feature space. Then, we perform the nearest neighbor classification with the negative Euclidean distance to predict the probability of query instance x_q belonging to category c by the following expression:

$$p\left(y_q = c \mid x_q\right) = \frac{\exp\left(-d\left(f_\phi\left(x_q\right), \mathbf{p}_c\right)\right)}{\sum_{c'=1}^{N} \exp\left(-d\left(f_\phi\left(x_q\right), \mathbf{p}_{c'}\right)\right)}, \forall x_q \in \mathcal{Q}_i, \tag{4}$$

where $d(\cdot, \cdot)$ denotes the Euclidean distance. Inspired by prior work [**?**], we apply a scale factor, γ, to adjust the similarity score, then the above equation becomes:

$$p\left(y_q = c \mid x_q\right) = \frac{\exp\left(-\gamma \cdot d\left(f_\phi\left(x_q\right), \mathbf{p}_c\right)\right)}{\sum_{c'=1}^{N} \exp\left(-\gamma \cdot d\left(f_\phi\left(x_q\right), \mathbf{p}_{c'}\right)\right)}, \forall x_q \in \mathcal{Q}_i. \tag{5}$$

During the experiments, we tune the initial values of the scale factor empirically and find it affects the meta-learning when the model is optimized based on pre-trained weights.

3.3. Dynamic kernel fusion network

We propose the Dynamic Kernel Fusion Network (DKF-Net), a simple yet effective embedding scheme for few-shot learning, to enrich the diversity and expressive capacity of typical backbones, e.g., Conv-4 [**? ? ?**], ResNet-12, and ResNet-18 [**?**]. DKF-Net aims to collect multi-scale spatial information by dynamically adjusting the receptive field size of neurons with Selective Kernel (SK) convolutions [**?**]. The top of Figure 4 depicts a SKUnit that is constituted of a $\{1 \times 1$ convolution, SK convolution, 1×1 convolution$\}$, and the bottom shows the complete DFK-net architecture.

The SK convolution performs dynamic fusion from multiple kernels via three operations - *Split*, *Fuse* and *Select*. Given a feature map $\mathbf{X} \in \mathbb{R}^{H' \times W' \times C'}$ with C' channels and spatial dimensions $H' \times W'$, as shown in Figure 4 (top part), we start by constructing two branches built upon transformations $\mathcal{F}_1$ and $\mathcal{F}_2$, mapping $\mathbf{X}$ to feature maps $\mathbf{U}_1 \in \mathbb{R}^{H \times W \times C}$ and $\mathbf{U}_2 \in \mathbb{R}^{H \times W \times C}$, separately. $\mathcal{F}_1$ and $\mathcal{F}_2$ refer to two convolutional operators with kernels 5×5 and 7×7, respectively, and are followed by Batch Normalization (BN) [**?**] and ReLU [**?**] in sequence. In practice, the $\mathcal{F}_2$ with a 7×7 kernel is displaced with a dilated convolutional layer with the rate of 2, which can alleviate further computational burden. This procedure is defined as *Split*.

We expect the neural network can adjust the RF sizes according to the stimulus content adaptively. An instinctive idea is to regulate the information flows from two branches by the second operation - *Fuse*. First, the two branches are initially integrated via element-wise summation, which can be expressed as:

$$\mathbf{U} = \mathbf{U}_1 + \mathbf{U}_2, \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ is the fused feature. Then, $\mathbf{U}$ is passed through a global average pooling (GAP) layer, which produces a channel-wise statistic $\mathbf{s} \in \mathbb{R}^{C}$ by shrinking feature

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

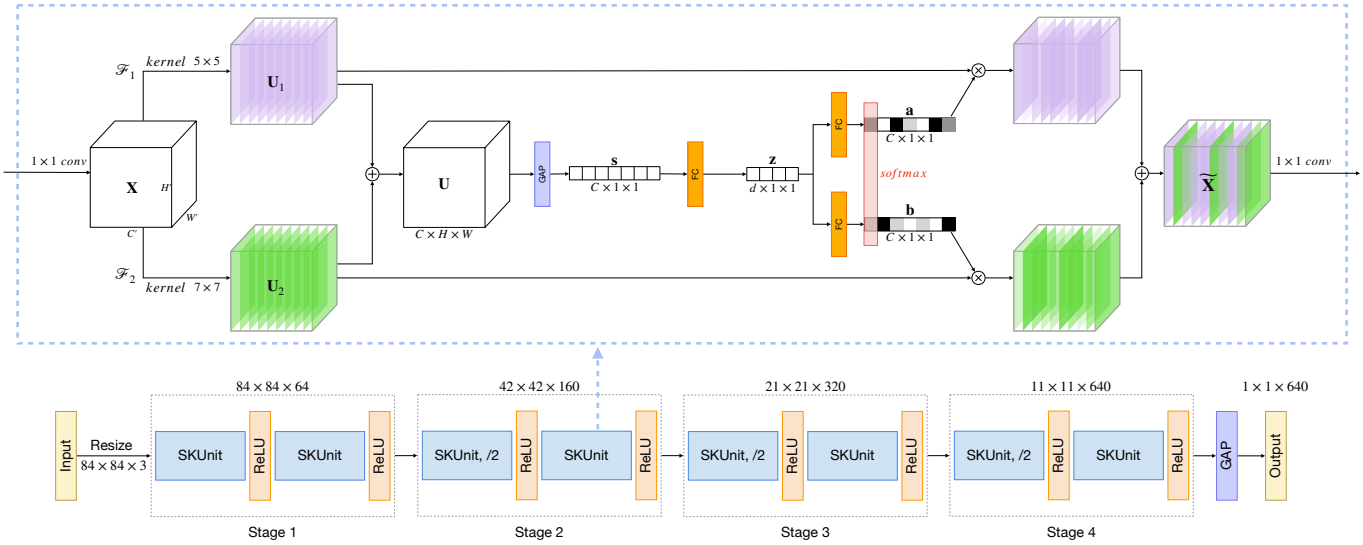

Figure 4. Schematic for the proposed Dynamic Kernel Fusion Network (DKF-Net).

maps through their spatial dimensions, $H \times W$. Formally, let s_c denote the c-th element of $\mathbf{s}$, it is calculated by:

$$s_c = \mathcal{F}_{GAP}(\mathbf{U}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j), \tag{7}$$

where $u_c(i,j)$ denotes the value at point (i,j) of the c-th channel $\mathbf{U}_c$. The vector $\mathbf{s}$ represents the importance of each channel, and it is further compressed to a compact feature descriptor $\mathbf{z} \in \mathbb{R}^d$ to save parameters and reduce dimensionality for better efficiency. Specifically, $\mathbf{z}$ is obtained by simply applying a fully connected (FC) layer to $\mathbf{s}$:

$$\mathbf{z} = \mathcal{F}_{FC}(\mathbf{s}) = \delta(\mathcal{B}(\mathbf{W}\mathbf{s})). \tag{8}$$

$\mathcal{F}_{FC}(\cdot)$ represents the fully connected operation defined by weights $\mathbf{W} \in \mathbb{R}^{d \times C}$, where $\mathcal{B}$ refers to the BN [**?**] and δ denotes the the ReLU [**?**] function. Thus, the number of channels is reduced to $d = \max((C/r), L)$, where r indicates the compression ratio and L is the minimum value of d. Following previous work [**? ?**], we empirically set r to 16 and L to 32.

Finally, the last operation - *Select*, guided by the compact feature descriptor $\mathbf{z}$, is applied to fulfill a dynamic adjustment of multi-scale spatial information. This is achieved by a control gate mechanism based on soft attention to assign the importance of each branch across channels. Specifically, let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^C$ be the soft attention vectors for $\mathbf{U}_1$ and $\mathbf{U}_2$; the channel-wise weights can be obtained by applying a softmax operator:

$$a_c = \frac{e^{\mathbf{A}_c \mathbf{z}}}{e^{\mathbf{A}_c \mathbf{z}} + e^{\mathbf{B}_c \mathbf{z}}}, b_c = \frac{e^{\mathbf{B}_c \mathbf{z}}}{e^{\mathbf{A}_c \mathbf{z}} + e^{\mathbf{B}_c \mathbf{z}}}, \tag{9}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{C \times d}$, $\mathbf{A}_c \in \mathbb{R}^{1 \times d}$ denotes the c-th row of $\mathbf{A}$ and a_c denotes the c-th element of $\mathbf{a}$; $\mathbf{B}_c$ and b_c are likewise. It is noteworthy that a_c and b_c have a relationship of $a_c + b_c = 1$ as there are only two branches in our case. We now have the refined feature map $\widetilde{\mathbf{X}}$ by applying the attention vectors $\mathbf{a}$ and $\mathbf{c}$ to each branch along the channel dimension:

$$\tilde{\mathbf{X}}_c = a_c \cdot \mathbf{U}_{1,c} + b_c \cdot \mathbf{U}_{2,c}, \tag{10}$$

where $\tilde{\mathbf{X}}_c$ refers to the c-th channel of $\widetilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}_c \in \mathbb{R}^{H \times W}$.

The proposed DFK-Net contains four stages with a block of {SK-Unit, ReLU, SK-Unit, ReLU} in each, as illustrated in Figure 4 (bottom part). We set the filters in each stage to 64,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

160, 320, 640, respectively, and add an 11×11 GAP layer after the last stage, which outputs 640-dimensional embeddings.

3.4. Embedding adaption via transformer

Up until now, the embedding function $f_\phi(\cdot)$, parameterized by ϕ, is assumed to be *task-agnostic*; we argue that such a setting is not ideal since the knowledge, i.e., the discriminative visual features learned in the base set, are equally effective to any novel categories. Here, we propose an embedding adaptation module that tailors the visual knowledge extracted from the base set, i.e., SEEN categories, into adaptive knowledge according to a specific task. We visualize this concept in Figure 5 schematically.

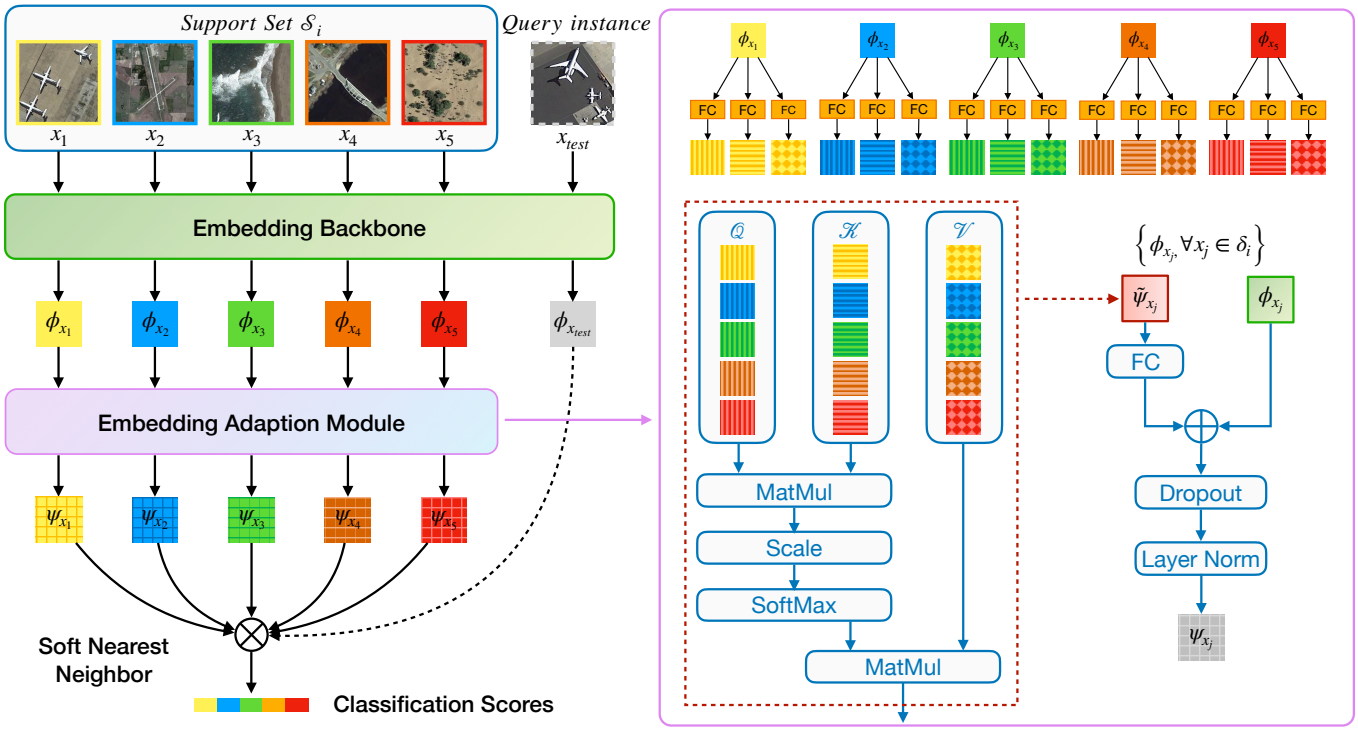

Figure 5. Illustration of the structure of our embedding adaption module, implemented with a set-to-set function based on Transformer. The right part shows the adaption step that transforms the embedding $f_\phi(x)$ to $f_\psi(x)$. For notation simplicity, we use ϕ_x and ψ_x instead of $f_\phi(x)$ and $f_\psi(x)$ in the figure, respectively.

Our embedding adaption module is achieved by contextualizing the instances over a set; thus, each of them has strong co-adaptation. Concretely, given a task-agnostic embedding function $f_\phi(x)$, let $\mathbf{T}$ denote a set-to-set function that transforms $f_\phi(x)$ to a *task-adaptive* embedding function $f_\psi(x)$. We treat the instances as bags or a set without order, requiring the set-to-set function $\mathbf{T}$ to output an adaptive set of instance embeddings while keeping permutation-invariant. The transformation step can be formalized in the following way:

$$\begin{aligned} \{f_\psi(x), \forall x \in \mathcal{S}_i\} &= \mathbf{T}(\{f_\phi(x); \forall x \in \mathcal{S}_i\}) \\ &= \mathbf{T}(\pi\{f_\phi(x); \forall x \in \mathcal{S}_i\}), \end{aligned} \tag{11}$$

where $\mathcal{S}_i$ is the support set of a target task, and $\pi(\cdot)$ is a permutation operator over a set that ensures the adapted embeddings will not change regardless of $\mathbf{T}$ receiving a set of input instances in any order. Inspired by the Transformer networks [**?**], we utilize dot-product self-attention to implement the set-to-set function $\mathbf{T}$. In the following, we use ϕ_x and ψ_x instead of $f_\phi(x)$ and $f_\psi(x)$ for the sake of notation simplicity.

Following the literature [**?**], we can describe the Transformer layer by defining the triplets $(\mathcal{Q}, \mathcal{K}, \mathcal{V})$ to indicate the set of the *queries*, *keys*, and *values*. Note that, in order to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

avoid the unfortunate double use of the term "query", we use italics to denote the "*query*" in the transformer layer to emphasize the difference from the "query set" in the few-shot tasks. Mathematically, for any instance x_j that belongs to $\mathcal{S}_i$, we first obtain its *query* by $\mathbf{q}_j = W_Q^\top \phi_{x_j}; \forall x_j \in \mathcal{S}_i$, where W_Q is a linear matrix. Similarly, the *key-value* pairs $\mathbf{k}_j$ and $\mathbf{v}_j$ are generated with W_K and W_V, respectively. For notion brevity, the bias in the linear projection is omitted here. Next, the similarity between an instance x_j with others in the support set can be measured by the scaled dot-product attention:

$$\tilde{\alpha}_{j,k} = \frac{\exp\left(\alpha_{j,k}\right)}{\sum_{k'} \exp\left(\alpha_{j,k'}\right)}, \alpha_{j,k} = \frac{\mathbf{q}_j^\top \mathbf{k}_{j,k}}{\sqrt{d}}; \quad \forall x_k \in \mathcal{S}_i, \tag{12}$$

where d is the dimensionality of the *queries* and *keys*. This similarity score then serves as weights for the transformed embedding of x_j:

$$\tilde{\psi}_{x_j} = \sum_k \tilde{\alpha}_{j,k} \mathbf{v}_k, \tag{13}$$

where $\mathbf{v}_k$ is the *value* of the k-th instance in $\mathcal{S}_i$. Finally, the *task-adaptive* embedding is given by:

$$\psi_{x_j} = \tau\left(\phi_{x_j} + W_{\mathbf{FC}}^\top \tilde{\psi}_{x_j}\right), \tag{14}$$

where $W_{\mathbf{FC}}$ indicates the projection weights of a fully connected layer and τ represents a procedure that further transforms the embedding by performing dropout [**?**] and layer normalization [**?**]. The whole flow of our Transformer module is illustrated in the right part of Figure 5.

4. Experimental Results

We verify the effectiveness of our proposed method Task-Adaptive Embedding Learning (TAEL) on two challenging datasets: NWPU-RESISC45 [15] and RSD46-WHU [22]. We will first introduce the datasets in Section 4.1 and then provide the implementation details in Section 4.2. Finally, we summarize the main results in Section 4.3.

4.1. Datasets

NWPU-RESISC45. The NWPU-RESISC45 dataset is a collection of remote sensing scene images extracted from Google Earth by experienced experts, proposed by Cheng et al. in 2017 [15]. It is composed of 45 categories within each category containing 700 images with a size of 256 × 256. In order to compare fairly with state-of-the-art (SOTA) algorithms for few-shot classification, we rely on the split setting proposed by Ravi et al. [39], and used in the prior FSL works [18**?** ,19] on the RS scene, which includes 25 categories for meta-training, 8 for meta-validation, and the rest 12 for meta-testing, as shown in Table 1. Specifically, the pre-training and meta-learning stages are performed on the 25 SEEN categories, and the best model is chosen based on the few-shot classification performance on the HELD-OUT meta-val split (UNSEEN). This serves as our final model, and it is evaluated on few-shot tasks sampled from the meta-test split (UNSEEN) without further fine-tuning. Following the most common setting in FSL [**? ? ?**], all images are first resized to 84 × 84 pixels.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

Table 1. NWPU-RESISC45 Dataset split.

Dataset-split		Categories
base (SEEN)	pre-training; meta-training	(1) Airplane; (2) Airport; (3) Baseball diamond; (4) Basketball court; (5) Beach; (6) Bridge; (7) Chaparral; (8) Church; (9) Circular farmland; (10) Cloud; (11) Commercial area; (12) Dense residential; (13) Desert; (14) Forest; (15) Freeway; (16) Golf course; (17) Ground track field; (18) Harbor; (19) Industrial area; (20) Intersection; (21) Island; (22) Lake; (23) Meadow; (24) Medium residential; (25) Mobile home park;
val (HELD-OUT)	meta-validation	(1) Mountain; (2) Overpass; (3) Palace; (4) Parking lot; (5) Railway; (6) Railway station; (7) Rectangular farmland; (8) River;
novel (UNSEEN)	meta-test	(1) Roundabout; (2) Runway; (3) Sea ice; (4) Ship; (5) Snowberg; (6) Sparse residential; (7) stadium; (8) Storage tank; (9) Tennis court; (10) Terrace; (11) Thermal power station; (12) Wetland;

RSD46-WHU Dataset split. The RSD46-WHU dataset is collected from Google Earth and Tianditu by hand, and released by Long et al [22]. It includes 46 categories, with around 500 – 3000 RS scene images in each and 117,000 in total. Similar to NWPU-RESISC45, we partition it into 26, 8, and 12 categories for meta-training, meta-validation, and meta-testing, respectively; see Table 2 for details. Likewise, all images are resized to 84×84 pixels.

Table 2. RSD46-WHU Dataset split.

Dataset-split		Categories
base (SEEN)	pre-training; meta-training	(1) Airplane; (2) Airport; (3) Artificial dense forest land; (4) Artificial sparse forest land; (5) Bare land; (6) Basketball court; (7) Blue structured factory building; (8) Building; (9) Construction site; (10) Cross river bridge; (11) Crossroads; (12) Dense tall building; (13) Dock; (14) Fish pond; (15) Footbridge; (16) Graff; (17) Grassland; (18) Low scattered building; (19) Lrregular farmland; (20) Medium density scattered building; (21) Medium density structured building; (22) Natural dense forest land; (23) Natural sparse forest land; (24) Oiltank; (25) Overpass; (26) Parking lot;
val (HELD-OUT)	meta-validation	(1) Plasticgreenhouse; (2) Playground; (3) Railway; (4) Red structured factory building; (5) Refinery; (6) Regular farmland; (7) Scattered blue roof factory building; (8) Scattered red roof factory building;
novel (UNSEEN)	meta-test	(1) Sewage plant-type-one; (2) Sewage plant-type-two; (3) Ship; (4) Solar power station; (5) Sparse residential area; (6) Square; (7) Steelsmelter; (8) Storage land; (9) Tennis court; (10) Thermal power plant; (11) Vegetable plot; (12) Water;

4.2. Implementation Details

We use the proposed DFK-Net as the embedding backbone for both the pre-training stage and meta-learning stage, and the architecture of DFK-Net is stated in Section 3.3.

Pre-training strategy. During the pre-training stage, the embedding backbone is trained as a typical classifier to classify all the categories in $\mathcal{D}_{\text{base}}$, e.g., 25 categories in NWPU-RESISC45, with CE loss. As MetaOptNet [**?**] suggested, the training is performed with data augmentation, i.e., random flip, crop, and color jittering, to increase the diversity

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

of training data. After each epoch, we sample 200 N_{val}-way 1-shot ($N_{val} = 8$) episodes from the meta-validation set $\mathcal{D}_{\text{val}}$. Then, the best pre-trained model is selected based on the average accuracy of 8-way 1-shot classification over the 200 episodes. Later on, the pre-trained weights of the best model are leveraged to initialize the embedding network and will be further optimized during the meta-learning stage.

Optimization. We train the model for 500 epochs in the pre-training stage and employ SGD with the Nesterov momentum of 0.9 for optimizing. The weight decay in SGD is 0.0005, and the initial learning rate is 0.001. We shrink the learning rate by 10 at 75, 150, and 300 epochs. In the meta-learning stage, we empirically meta-train the model for 200 epochs, and each contains 100 N-way K-shot episodes/tasks. Each mini-batch contains 16 tasks during training. The initial learning rate in this stage is set to 0.0001, while the other parameters in SGD are the same as in pre-training. We empirically tune the scale factor γ in Equation (5) from the reciprocal of $\{0.1, 1, 10, 16, 32, 64\}$ and find 64 is the best. Furthermore, for the Embedding Adaption Module, the dropout rate in the transformer is set to 0.5.

4.3. Main Results

We have evaluated our method on two challenging RS scene datasets, namely NWPU-RESISC45 [15] and RSD46-WHU [22]. Following [18**? ?**], the standard evaluation protocols are used in all our experiments, exactly as in corresponding compared works. All the experiments are constructed and evaluated on the most commonly used 5-way 1-shot and 5-way 5-shot classification settings. Of note, in these experiments, keeping with the spirit that training and testing conditions should be consistent, the task configuration for meta-training, meta-validation, and meta-testing is the same. For instance, consider the 5-way 1-shot scenario. A 5-way 1-shot task is composed of 5 random sampled categories, and each category includes 1 support instance and 15 unlabeled query instances, which are used for training and inference, respectively. We keep sampling 5-way 1-shot tasks from the base set during the meta-training phase and set 100 tasks as an epoch. Then, at the end of each epoch, we feed 600 tasks drawn from the HELD-OUT validation set to the model and record the 5-way 1-shot classification accuracy. We train the meta-learner for 200 epochs and select the best one based on the 5-way 1-shot classification performance on the validation set.

As depicted in Figure 6 (a), left pane, we can see that the best model of 5-way 1-shot on NWPU-RESISC45 appeared at the 177-th epoch, with a correspondingly low loss. In addition, we do not utilize any data from the meta-test set during training nor perform further fine-tuning during meta-testing. Once the meta-training procedure is done, the performance of the proposed method TAEL is finally evaluated by the mean accuracy over 10,000 5way-1shot tasks randomly sampled from the meta-testing split, with 95% confidence intervals, and the same goes for the 5-way 5-shot scenario. Note that most previous approaches [**? ? ? ? ?**] are evaluated with 600-2000 tasks sampled from the meta-testing split according to their original setup, which introduce high variance, as shown in Tables 3 and 4. We adhere to one key principle that avoids falsely embellishing the capabilities of our method by overfitting a specific dataset. That is, in all experiments, whether 1-shot or 5-shot, as described in Section 4.2, we keep all the hyper-parameters in the pre-training and meta-learning stages the same for both datasets. Figure 6 (b) shows that the performances of TAEL are not so steady on the meta-validation split of RSD46-WHU. This is probably on account of the RSD46-WHU dataset containing lower quality images, which is extremely challenging for the severe low-data scenarios.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

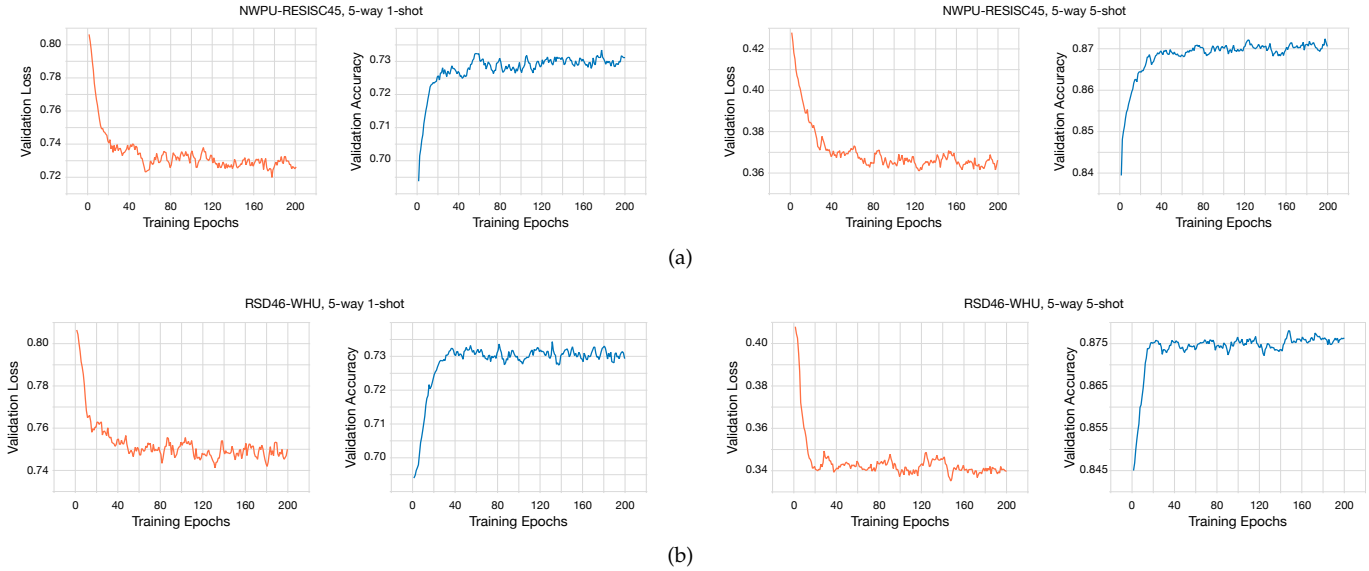

Figure 6. Meta-validation accuracy and loss curves for TAEL on (a) NWPU-RESISC45 5way-1shot and 5way-5shot, and (b) RSD46-WHU 5way-1shot and 5way-5shot. All curves are smoothed with a rate of 0.8 for better visualization.

The few-shot classification accuracies on NWPU-RESISC45 and RSD46-WHU for TAEL and other previous methods are summarized in Tables 3 and 4, respectively. Methods with * indicate that the original backbone of which has been replaced by ResNet-12, and corresponding results are reported in [18]. As seen in Tables 3 and 4, we can see that the proposed TAEL is uniformly better than SOTA algorithms on both 1-shot and 5-shot regimes for the NWPU-RESISC45 and RSD46-WHU datasets. By jointly leveraging the strengths of multi-scale kernel fusion and task-adaptive embedding learning, TAEL improves over the RS scene few-shot classification baseline [18] across all datasets by approximately 2-4% for both 1-shot and 5-shot scenarios. We can also observe from Table 4 that our method TAEL outperforms the current best results (RS-SSKD [**?**]) on NWPU-RESISC45 by 2.55% in the 1-shot task, whereas for the 5-shot task, it improves the accuracy by 0.94%. For the RSD46-WHU dataset, Table 4 displays TAEL surpass RS-SSKD by 1.54% and 1.84% for 1 and 5-shot, respectively.

Table 3. Comparison to previous works on NWPU-RESISC45. Average 5-way few-shot classification accuracy (%) is reported with 95% confidence intervals. The symbol * denotes the backbone of the original model is replaced with Resnet-12, and the results are reported in [18]. The best results in each column are marked in bold.

Method	Backbone	1-shot	5-shot
ProtoNet [**?**]	Conv-4	51.17 0.79	74.58 0.56
ProtoNet*	ResNet-12	62.78 0.85	80.19 0.52
MAML [**?**]	Conv-4	53.52 0.83	71.69 0.63
MAML*	ResNet-12	56.01 0.87	72.94 0.63
RelationNet [**?**]	Conv-4	57.10 0.89	73.55 0.56
RelationNet*	ResNet-12	55.84 0.88	75.78 0.57
TADAM [**?**]	ResNet-12	62.25 0.79	82.36 0.54
MetaOptNet [**?**]	ResNet-12	62.72 0.64	80.41 0.41
DSN-MR [**?**]	ResNet-12	66.93 0.51	81.67 0.49
FEAT [**?**]	ResNet-12	68.27 0.19	83.51 0.11
Zhang et al. [18]	ResNet-12	69.46 0.22	84.66 0.12
RS-SSKD [**?**]	ResNet-12	70.64 0.22	86.26 0.12
TAEL (ours)	DKF-Net	**73.19 0.19**	**87.20 0.10**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

Table 4. Comparison to previous works on RSD46-WHU. Average 5-way few-shot classification accuracy (%) is reported with 95% confidence intervals. The symbol * denotes the backbone of the original model is replaced with Resnet-12, and the results are reported in [18].The best results in each column are marked in bold.

Method	Backbone	1-shot	5-shot
ProtoNet [?]	Conv-4	52.57 0.89	71.95 0.71
ProtoNet*	ResNet-12	60.53 0.99	77.53 0.73
MAML [?]	Conv-4	52.73 0.91	69.18 0.73
MAML*	ResNet-12	54.36 1.04	69.28 0.81
RelationNet [?]	Conv-4	55.18 0.90	68.86 0.71
RelationNet*	ResNet-12	53.73 0.95	69.98 0.74
TADAM [?]	ResNet-12	65.84 0.67	82.79 0.58
MetaOptNet [?]	ResNet-12	62.05 0.76	82.60 0.46
DSN-MR [?]	ResNet-12	66.53 0.70	82.74 0.54
FEAT [?]	ResNet-12	71.04 0.21	85.27 0.13
Zhang et al. [18]	ResNet-12	69.08 0.25	84.10 0.15
RS-SSKD [?]	ResNet-12	71.73 0.25	85.90 0.15
TAEL (ours)	DKF-Net	**73.27 0.20**	**87.74 0.12**

To compare whether the embedding backbone impacts the performance of FSL algorithms, we plot bar charts in Figures 7 and 8 for a better observation. The bars with dots denote the re-implementation of methods in which the original backbone are replaced by ResNet-12 [?], and the results are provided by [18]. Surprisingly, Figures 7 and 8 show that the re-implementation of MAML [?] gets notable improvements over the Conv-4 version on the 1-shot scenario of both datasets, whereas only minor improvement is obtained in the 5-shot case on RSD46-WHU. For ProtoNet [?] equipped with ResNet-12, even bigger improvements are achieved on both datasets, especially in the 1-shot case of NWPU-RESISC45, which improves 11.61%. On the contrary, Relation-Net [?] becomes even worse in the 1-shot setting for both datasets, which may be due to its auxiliary comparison module leading to over-fitting when using deeper networks. Generally speaking, the gap among different approaches drastically diminishes when the backbone goes deeper.

MAML claims that using a deeper backbone rather than Conv-4 may cause overfitting; however, this issue is overcome by applying data augmentation such as random crop, horizontal flip, and color jitter suggested in MetaOptNet [?]. Such data augmentation has become a standard operation in current methods [18? ? ? ?]. The rest of the comparison methods in this work, including our own, are built upon ProtoNet. DSN-MR [?] achieves considerable performance while consuming many computational resources since the similarity measure is performed in subspaces. The work [18], FEAT [?], and RS-SSKD [?] demonstrate that learning better embeddings can significantly improve performance. TADAM [?] attempts to retrieve task-specific embeddings for each target task based on an additional task-conditioning module. The authors of TADAM adopt an auxiliary co-training strategy to alleviate the computational burden, yet extra parameters and additional complexity are introduced to the network. As with TADAM, the embedding adaption module in our method also provides task-adaptive embeddings that generate discriminative embeddings tailored to target tasks while coming at a modest increase in computational cost. We perform further analysis on training time and computational cost of all comparative methods and ours in Section 5.3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

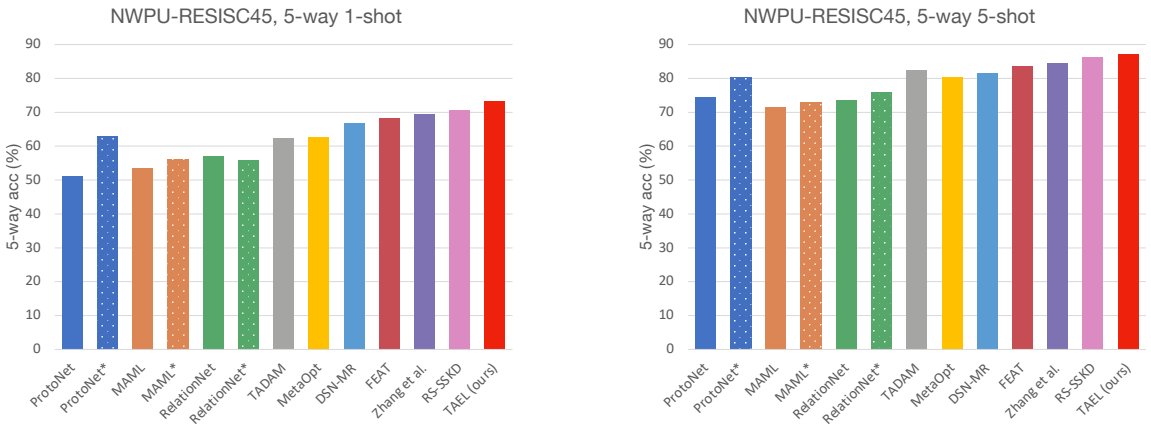

Figure 7. The few-shot classification performance (with 95% confidence intervals) on the NWPU-RESISC45 dataset, the bars with dots indicate the re-implementation of approaches with Resnet-12 backbone.

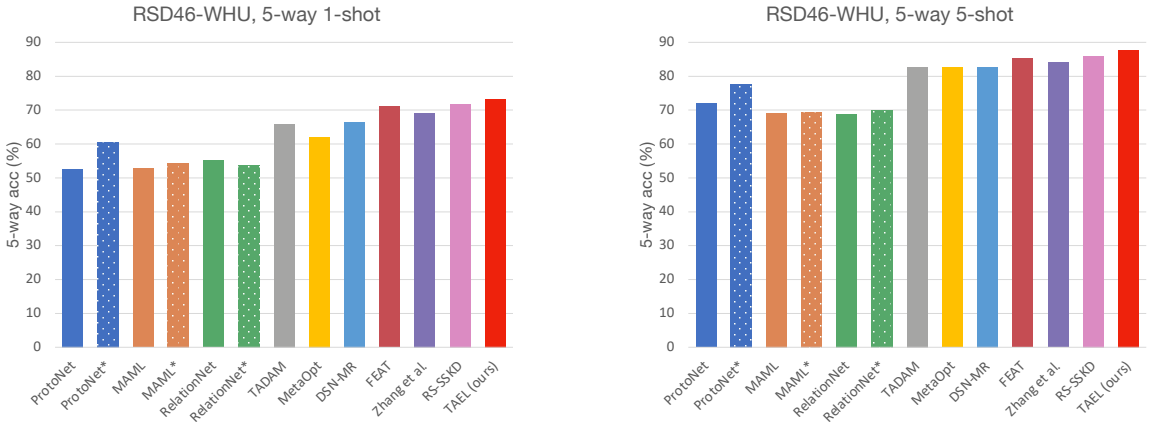

Figure 8. The few-shot classification performance (with 95% confidence intervals) on the RSD46-WHU dataset, the bars with dots indicate the re-implementation of approaches with Resnet-12 backbone.

5. Discussion

5.1. Effect of Different Embedding Networks

To verify the effectiveness of the proposed embedding network DFK-Net in our method, we perform an ablation study on the NWPU-RESISC45 and RSD46-WHU datasets by changing the embedding backbone to the most popular architectures in few-shot learning, i.e., the 4-layer convolution network (Conv-4) adopted in [**? ? ?**] and the 12-layer residual network (ResNet-12) adopted in [18**? ? ? ? ?**]. We use Adam [**?**] and SGD to optimize the Conv-4 and ResNet-12 variants, respectively.

The Conv-4 network is constituted by four repeated blocks. Each block is a sequential concatenation of {3×3 convolution with k filters, batch normalization, ReLU, and max-pooling with size 2}. The number of filters in each block is set to 64; namely, the network architecture is 64-64-64-64, the same as in ProtoNet [**?**]. We apply a global max-pooling layer with size 5 after the last block to reduce the computational cost.

We employ the ResNet-12 structure as suggested in [**?**], which contains four residual blocks, each of which repeats the following convolutional block three times {3×3 convolution with k filters, batch normalization, Leaky ReLU(0.1)}. Then a 2×2 max-pooling layer with stride 2 is applied at the end of each residual block. The number of filters k starts with 64 and is then set to 160, 320, 640, respectively. At last, we apply a 5×5 global average pooling (GAP) layer, which generates 640-dimensional embeddings.

The architecture of the proposed DFK-Net is stated in Section 3.3, and please see Figure 4. In addition, we have further experimented with a DFK-Net variant, denoted

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

as DFK-Net †, by changing the filters in each stage to 64, 256, 512, and 1024 respectively, which yields 1024-dimensional embeddings.

Figure 9 shows the few-shot classification results of our model with different embedding networks, including the Conv-4, ResNet-12, the proposed DFK-Net, and the DFK-Net variant. Results on the NWPU-RESISC45 dataset show a clear tendency that the performance gap among different backbones significantly reduces when the embedding architecture gets deeper. On the RSD46-WHU dataset, a similar trend can be observed. Moreover, we can also observe that our model using DFK-Net consistently outperforms the ablation using ResNet-12 on both datasets with a margin, which indicates that the proposed embedding backbone is very efficient. We attribute the success of DFK-Net for few-shot classification to two factors: its ability to dynamically weight the averaging of features from multiple kernels according to the receptive field size while its high parameter efficiency is suited well with the low data regime.

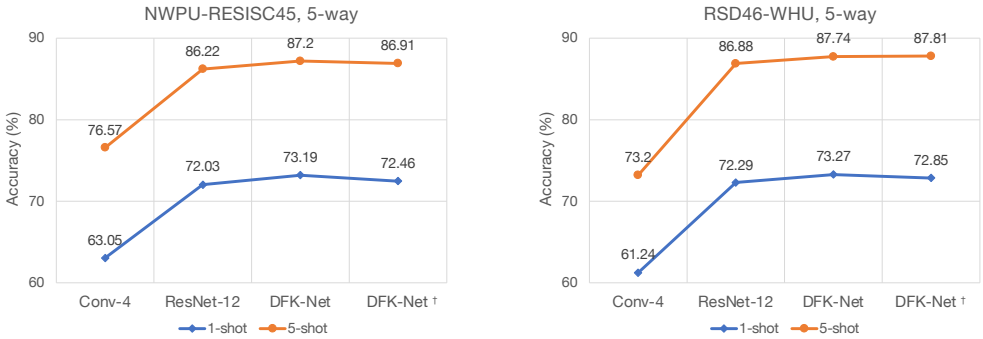

Figure 9. Few-shot classification accuracy of our model using different embedding networks on the NWPU-RESISC45 and RSD46-WHU datasets.

Table 5 reports the number of parameters and FLOPs [**?**] of each embedding network. We can see that the Conv-4 has quite a low amount of parameters and FLOPs, whereas it degrades the accuracy of our model a lot. We conjecture that the shallow architecture of Conv-4 is responsible for the failure of the performance as it does not adequately use our model's expressive capacity and leads to underfitting. As illustrated in Table 5, the number of parameters and FLOPs of DFK-Net is slightly more than half of ResNet-12 due to the grouped and dilated convolutions adopted in our architecture. For further comparison, we conduct a variant of DFK-Net by changing its depth to match the complexity of ResNet-12, denoted as DFK-Net †. Surprisingly, Figure 9 shows that DFK-Net †, i.e., the increased complexity version, does not lead to better accuracy with respect to the original DFK-Net, except in the 5-way 5-shot scenario of RSD46-WHU. A potential explanation is that the optimization process of meta-learner becomes more difficult with so few data points when increasing the number of parameters and the size of backbone, which trends to overfitting. It is crucial to find a trade-off between the model's generalization capacity and parameter efficiency. All above, we conclude that the advantage of DFK-Net can be attributed to the adaptive fusion mechanism of weighted multi-scale information from different kernels and a high parameter efficiency, which yields more diversity and better generalization ability for few-shot classification.

Table 5. Parameter efficiency of different embedding networks. #P stands for the number of parameters, and FLOPs denotes the number of multiply-adds, following the definition in [**?**]. DFK-Net † is a variant of the proposed embedding network.

Backbone →	**Conv-4**	**ResNet-12**	**DFK-Net**	**DFK-Net †**
#P	0.11M	12.42M	6.25M	11.44M
FLOPs	98.96M	3518.31M	1903.37M	3480.50M

 doi:10.20944/preprints202108.0389.v1

5.2. Effect of Embedding Adaption Module

To investigate whether the embedding adaption module is indeed effective, we perform analyses for our method TAEL and its ablated variants on both datasets: NWPU-RESISC45 and RSD46-WHU. The following experiments are established on the proposed embedding backbone DFK-Net.

We start by evaluating our method with and without the embedding adaption module. As stated in Section 3.4, if we train a model without embedding adaption, the embedding function is assumed to be *task-agnostic*, and we name this vanilla model as *ours-agnostic*. Then, we apply the embedding adaption procedure to the data in the support set to construct the classifier (see figure 5). In this case, the extracted visual knowledge will be transformed into task-adaptive knowledge according to a specific task and yielding more discriminative embeddings; thus, we name this model *ours-adaptive*, i.e., the proposed method TAEL. As seen from table 6, the model using task-adaptive embeddings achieves better performance than the vanilla model, especially in the 1-shot scenario, which gains an approximately 2%-3% promotion. This confirms that the proposed embedding adaption module can efficiently tailor the common embeddings to task-adaptive embeddings, which are more discriminative to a specific target task. These experimental results support our hypothesis: embedding is one of the most crucial factors in few-shot learning, and we can expect that better embeddings lead to better FSL performance.

Table 6. Alation study of whether the embedding adaption module improves the performance of few-shot classification. Results are averaged over 10,000 test tasks with 95% confidence intervals.

Model	NWPU-RESISC45, 5-way		RSD46-WHU, 5-way	
	1-shot	5-shot	1-shot	5-shot
Ours-agnostic	70.87 0.19	85.62 0.11	70.11 0.21	85.91 0.13
Ours-adaptive	73.19 0.19	87.20 0.10	73.27 0.20	87.74 0.12

We further investigate the impact of different architectural choices of the Transformer in our embedding adaption module. In our current TAEL model, the embedding adaption is implemented by a set-to-set function with Transformer [**?**], in which we adopt a shallow architecture of simply one attention head and one layer. We follow the common practice in [**?**] to conduct the Transformer with more complex structures, e.g., multiple heads and deeper stacked layers. First, we replace the single head attention in our module with multi-head attention by increasing the number of heads to 2, 4, and 8 while fixing the number of layers to one. The performance of one-head and multi-head ablations on 5-way classification are summarized in Table 7. The results indicate that the multi-head ablations provide minimal benefits or even harm the performance while introducing extra computational costs. Fixing the attention head to one, we next turn to stack the layers in the Transformer to 2 and 3. From Table 8, we see barely any improvements for this change, and in fact, the performance often drops with respect to the one-layer structure. Thus, we empirically speculate that, under an extremely low data regime like few-shot learning, complex structures do not always result in performance promotion since the difficulty of optimization also increases, and the model becomes more difficult to converge.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

Table 7. Ablation study on the number of attention heads in the proposed embedding adaption module. Results are averaged over 10,000 test tasks with 95% confidence intervals.

#Heads	NWPU-RESISC45, 5-way		RSD46-WHU, 5-way	
	1-shot	5-shot	1-shot	5-shot
1	73.19 0.19	87.20 0.10	73.27 0.20	87.74 0.12
2	72.77 0.20	87.09 0.10	73.02 0.20	87.82 0.12
4	73.28 0.19	87.14 0.10	72.93 0.20	87.27 0.12
8	73.12 0.19	87.26 0.10	72.50 0.20	87.56 0.12

Table 8. Ablation study on the number of layers in Transformer of the proposed embedding adaption module. Results are averaged over 10,000 test tasks with 95% confidence intervals

#Layers	NWPU-RESISC45, 5-way		RSD46-WHU, 5-way	
	1-shot	5-shot	1-shot	5-shot
1	73.19 0.19	87.20 0.10	73.27 0.20	87.74 0.12
2	73.68 0.19	87.04 0.10	73.15 0.20	87.81 0.12
3	72.93 0.19	87.25 0.10	72.84 0.20	87.40 0.12

5.3. Training Time Analysis

In this section, we compare the meta-training time of our model TAEL with the state-of-the-art methods. We report the 5-way 1-shot and 5-way 5-shot runtime on both datasets: NWPU-RESISC45 and RSD46-WHU. To ensure a fair comparison, the prior works and ours are processed in the same experimental condition, i.e., AMD 2950X with 16 cores, 128GB RAM, and a single GPU GeForce RTX 3090. The only exception is the test for DSN-MR [**?**], which requires 2 RTX 3090GPUs, due to the high GPU memory consumption of its SVD step.

The Conv-4 adopted in ProtoNet [**?**], MAML [**?**], and RelationNet [**?**] differ in the number of filters per layer, which are 64-64-64-64, 32-32-32-32, and 64-96-128-256, respectively. The architectures of ResNet-12 and the proposed DFK-Net are stated in Section 5.1. In practice, the meta-training time is heavily dependent on the number of epochs and how many N-way K-shot episodes/tasks are in each epoch, which is set empirically by the authors. Our tests of all methods follow their original settings. For example, the early FSL methods like ProtoNet, MAML, and RealationNet are trained with 600 epochs, each containing 100 tasks. MetaOptNet [**?**] suggests training with fewer epochs while setting more tasks per epoch, e.g., 1000 tasks/epoch and 60 epochs in total. Most current methods followed the latter setting, e.g., Zhang et al. [18] and RS-SSKD [**?**] set each epoch with 800 tasks and meta-trained for 60 epochs. Our model follows the setting of ProtoNet, where each epoch contains 100 tasks, yet only 200 epochs are meta-trained as we have an additional pre-training stage, enabling the model to converge faster in the meta-training phase. Table 9 summarizes the running times of the discussed methods on both datasets with respect to the number of total meta-training iterations.

From Table 9, we observe that the running time of ProtoNet and RelationNet only slightly increase when the backbone is changed to Resnet-12. The evaluation of MAML is performed on its first-order approximation version by ignoring second-order derivatives to speed up the training time. However, MAML using Resnet-12 as the backbone still increases the running time by more than two times that of the original version using Conv-4. We notice that the metric-based methods built upon ProtoNet, e.g., FEAT [**?**], Zhang et al. [18], and RS-SSKD [**?**], generally come with short training times. In comparison, while MetaOptNet [**?**] achieves a competitive performance of few-shot classification, its training time is significantly increased because it incorporates the differentiable quadratic programming solver to learn an end-to-end model with a linear classifier SVM. DSN-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 18 August 2021 doi:10.20944/preprints202108.0389.v1

Table 9. Meta-training runtime comparison on NWPU-RESISC45 and RSD46-WHU datasets, under 5-way 1-shot and 5-way 5-shot classification scenarios.

Method	Backbone	Meta-training iterations	NWPU-RESISC45		RSD46-WHU	
			1-shot runtime	5-shot runtime	1-shot runtime	5-shot runtime
ProtoNet [?]	Conv-4	60,000	1.2 h	1.4 h	1.2 h	1.4 h
ProtoNet *	ResNet-12	60,000	1.8 h	1.9 h	1.8 h	1.9 h
MAML [?]	Conv-4	60,000	7.7 h	8.3 h	7.8 h	8.3 h
MAML*	ResNet-12	60,000	18.2 h	19.5 h	18 h	19.5 h
RelationNet [?]	Conv-4	60,000	1.4 h	1.8 h	1.4 h	1.7 h
RelationNet *	ResNet-12	60,000	2.2 h	2.5 h	2.2 h	2.3 h
TADAM [?]	ResNet-12	30,000	5.9 h	7.5 h	7.4 h	9.5 h
MetaOpt [?]	ResNet-12	60,000	6.4 h	10.2 h	6.3 h	10.1 h
DSN-MR [?]	ResNet-12	80,000	33.2 h	70.3 h	32.9 h	70.5 h
FEAT [?]	ResNet-12	36,000	3.8 h	4.3 h	3.8 h	4.3 h
Zhang et al. [18]	ResNet-12	48,000	2.3 h	2.9 h	2.3 h	2.9 h
RS-SSKD [?]	ResNet-12	48,000	2.3 h	2.9 h	2.3 h	3.0 h
TAEL (ours)	DFK-Net	20,000	3.1 h	3.8 h	3.1 h	3.9 h

MR [?] constructs the classifier on closed-formed projection distance in subspaces. It is not surprising that DSN-MR has a very slow training time since its subspaces are obtained through a singular value decomposition (SVD) step, which is computationally expensive. Thanks to the embedded adaption module, our model converges quickly in the meta-training stage and needs fewer total training iterations (episodes/tasks) than other methods. As expected, the results show that our model is practical and offers absolute gains over the previous methods at a modest training time.

Additionally, we see an interesting phenomenon: almost all methods have virtually the same meta-training time on both datasets. The reason is simple, the running time per iteration is inherent for each method; thus, the meta-learning time depends on the total number of training iterations, which is the same on both datasets. The only exception is TADAM [?], which utilizes an auxiliary co-training scheme in the meta-training phase. This co-training scheme comes with a high computational cost due to introducing an additional logits head, i.e., the traditional M-way classification on base set where M is the number of all categories. We can easily infer that the running time of TADAM differs on the two datasets is because the burden of co-training consumes more on RSD46-WHU as it is larger than NWPU-RESISC4.

6. Conclusion

This work suggests that embedding is critical to few-shot classification as it plays dual roles - representing images and building classifiers in the embedding space. To this end, we have proposed a framework for a few-shot classification that complements the existing methods by refining the embeddings from two perspectives: a lightweight embedding network that fuses multi-scale information and a task-adaptive strategy that further tailors the embeddings. The former enriches the diversity and expressive capacity of embeddings by dynamically weighting information from multiple kernels, while the latter learns discriminative representations by transforming the universal embeddings into task-adaptive embeddings via a self-attention mechanism. We extensively evaluate our model, TAEL, on two datasets: NWPU-RESISC4 and RSD46-WHU. As shown in the results, TAEL outperforms current state-of-the-art methods and comes at a modest training time. Furthermore, the experimental results and ablation studies have verified our assumption that good embeddings positively affect the few-shot classification performance. While our method is effective and the experimental results are encouraging, much work can be done to achieve human-level performance in the low data regime. Our potential future work involves developing algorithms that suit extended settings, e.g., cross-dataset, cross-domain, transductive, and generalized few-shot learning. Another exciting direction for

our future work is to extend the standard few-shot learning to a continual setting in which training and testing stages do not have to be separated, and instead, models are evaluated while learning novel concepts as in the real world.

Author Contributions: Conceptualization, P.Z., G.F., and D.W.; Data curation, P.Z., C.W., and D.W.; Investigation, P.Z., C.W., and D.W.; Methodology, P.Z., and D.W.; Software, P.Z, C.W., and D.W.; Validation, P.Z., and C.W., visualization, P.Z.; Writing — original draft, P.Z.; Writing — review & editing, P.Z. and G.F.; Supervision, G.F., and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61871460; in part by the Shaanxi Provincial Key Research and Development Program under Grant 2020KW-003; in part by the Fundamental Research Funds for the Central Universities under Grant 3102019ghxm016.

Conflicts of Interest: The authors declare no competing financial interests. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; and in the decision to publish the results.

References

1. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 3965–3981.
2. Negrel, R.; Picard, D.; Gosselin, P.H. Evaluation of second-order visual features for land-use classification. 2014 12th International Workshop on Content-Based Multimedia Indexing (CBMI). IEEE, 2014, pp. 1–5.
3. Cheng, G.; Guo, L.; Zhao, T.; Han, J.; Li, H.; Fang, J. Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA. *International Journal of Remote Sensing* **2013**, *34*, 45–59.
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. On the use of unmanned aerial systems for environmental monitoring. *Remote sensing* **2018**, *10*, 641.
5. 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. *Landscape and Urban Planning* **2011**, *100*, 223–230.
6. El Garouani, A.; Mulla, D.J.; El Garouani, S.; Knight, J. Analysis of urban growth and sprawl from remote sensing data: Case of Fez, Morocco. *International Journal of Sustainable Built Environment* **2017**, *6*, 160–169.
7. Hansen, M.C.; Potapov, P.V.; Moore, R.; Hancher, M.; Turubanova, S.A.; Tyukavina, A.; Thau, D.; Stehman, S.; Goetz, S.J.; Loveland, T.R.; others. High-resolution global maps of 21st-century forest cover change. *science* **2013**, *342*, 850–853.
8. 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 transactions on geoscience and remote sensing* **2018**, *56*, 2811–2821.
9. Browne, D.; Giering, M.; others. PulseNetOne: Fast Unsupervised Pruning of Convolutional Neural Networks for Remote Sensing. *Remote Sensing* **2020**, *12*, 1092.
10. Kang, J.; Fernandez-Beltran, R.; Ye, Z.; Tong, X.; Ghamisi, P.; Plaza, A. Deep Metric Learning Based on Scalable Neighborhood Components for Remote Sensing Scene Characterization. *IEEE Transactions on Geoscience and Remote Sensing* **2020**.
11. Yu, D.; Xu, Q.; Guo, H.; Zhao, C.; Lin, Y.; Li, D. An Efficient and Lightweight Convolutional Neural Network for Remote Sensing Image Scene Classification. *Sensors* **2020**, *20*, 1999.
12. Wang, D.; Bai, Y.; Bai, B.; Wu, C.; Li, Y. Heterogeneous two-Stream Network with Hierarchical Feature Prefusion for Multispectral Pan-Sharpening. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021. IEEE, 2021, pp. 1845–1849.
13. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, 2010, pp. 270–279.
14. Xia, G.S.; Yang, W.; Delon, J.; Gousseau, Y.; Sun, H.; Maître, H. Structural High-resolution Satellite Image Indexing. ISPRS TC VII Symposium - 100 Years ISPRS; W., W.; Székely.; B.., Eds.; , 2010; Vol. XXXVIII, pp. 298–303.
15. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE* **2017**, *105*, 1865–1883.
16. Rußwurm, M.; Wang, S.; Körner, M.; Lobell, D. Meta-Learning for Few-Shot Land Cover Classification. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 788–796. doi:10.1109/CVPRW50498.2020.00108.
17. Liu, B.; Yu, X.; Yu, A.; Zhang, P.; Wan, G.; Wang, R. Deep few-shot learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* **2018**, *57*, 2290–2304.
18. Zhang, P.; Li, Y.; Wang, D.; Bai, Y.; Bai, B. Few-shot Classification of Aerial Scene Images via Meta-learning. *Remote Sensing* **2021**, *13*, 108.
19. Zhang, P.; Bai, Y.; Wang, D.; Bai, B.; Li, Y. A Meta-Learning Framework for Few-Shot Classification of Remote Sensing Scene. IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021. IEEE, 2021, pp. 4590–4594.

20. 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 preprint arXiv:2009.13364* **2020**.
21. Li, L.; Han, J.; Yao, X.; Cheng, G.; Guo, L. DLA-MatchNet for few-shot remote sensing image scene classification. *IEEE Transactions on Geoscience and Remote Sensing* **2020**.
22. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 2486–2498.