EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems Research Article EAI.EU

ERKT-Net: Implementing Efficient and Robust Knowledge Distillation for Remote Sensing Image Classification

Huaxiang Song[1,*], Yafang Li, Xiaowen Li, Yuxuan Zhang, Yangyan Zhu, and Yong Zhou

School of Geography Science and Tourism, Hunan University of Arts and Science, Changde, Hunan 415000, China

Abstract

The classification of remote sensing images (RSIs) poses a significant challenge due to the presence of clustered ground objects and noisy backgrounds. While many approaches rely on scaling models to enhance accuracy, the deployment of RSI classifiers often requires substantial computational resources, thus necessitating the use of lightweight algorithms. In this paper, we present an efficient and robust knowledge transfer network named ERKT-Net, which is designed to provide a lightweight yet accurate convolutional neural network (CNN) classifier. This method utilizes innovative yet straightforward concepts to better accommodate the inherent nature of RSIs, thereby significantly improving the efficiency and robustness of traditional knowledge distillation (KD) techniques developed on ImageNet-1K. We evaluate ERKT-Net on three benchmark RSI datasets. The results demonstrate that our model presents superior accuracy and a very compact size compared to 40 other advanced methods published between 2020 and 2023. On the most challenging NWPU45 dataset, ERKT-Net outperformed other KD-based methods with a maximum overall accuracy (OA) value of 22.4%. Using the same criterion, it also surpassed the first-ranked multi-model method with a minimum OA value of 0.6 but presented at least a 95% reduction in parameters. Furthermore, ablation experiments indicated that our training approach has significantly improved the efficiency and robustness of classic DA techniques. Notably, it can reduce the time expenditure in the distillation phase by at least 80%, with a slight sacrifice in accuracy. This study confirmed that a logit-based KD technique can be more efficient and effective in developing lightweight yet accurate classifiers, especially when the method is tailored to the inherent characteristics of RSIs.

Keywords: ERKT-Net, Variance-Suppression Strategy, Knowledge Distillation, Remote Sensing Image Classification, Deep Learning

Received on 3 January 2024, accepted on 17 May 2024, published on 3 July 2024

doi: 10.4108/eetinis.v11i3.4748

*Corresponding author. Email: cn11028719@huas.edu.cn

1. Introduction

Remote sensing images (RSIs) are pivotal data from Earth observation. As the capabilities of onboard devices expand, the volume of RSIs has grown so large that only computer algorithms can interpret RSIs efficiently [1]. Among these algorithms, classification is the most fundamental. A decade ago, shallow models dominated in RSI classification, necessitating substantial feature engineering knowledge in machine learning [2–3]. However, with the advent of deep learning, Convolutional Neural Networks (CNNs) [4–5] or Vision Transformers (ViTs) [6–7] have become the primary solutions for RSI classification, owing to their superiority in automatic feature extraction.

In recent time, a multitude of CNN- or ViT-based methods have emerged [8–9], demonstrating state-of-the-art (SOTA) performance for RSI classification. However, most of these advanced algorithms heavily rely on computational and storage resources to achieve competitive accuracy [10], which poses significant challenges for deployment in remote sensing tasks. For instance, applications such as in-orbit data

retrieval [11], real-time insect monitoring [12], or environmental surveys utilizing edge computing [13]

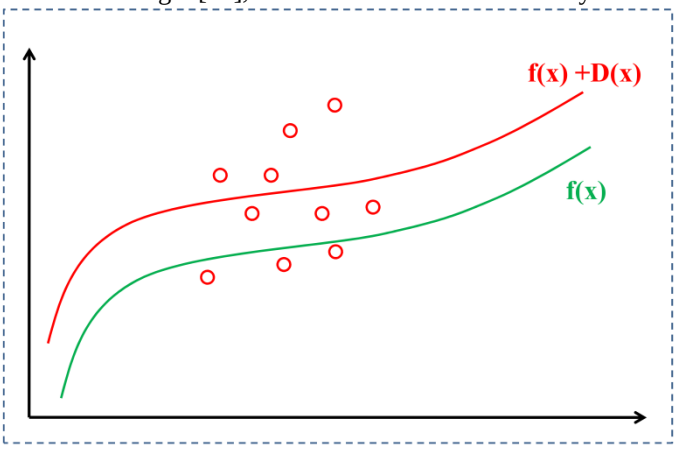

Figure 1. The impact of data distribution variance on function approximation.

commonly impose stringent hardware restrictions, which conflict with resource-intensive models. Therefore, a lightweight model with exceptional generalization capability is very important in the domain of RSI classification.

Pre-training on large-scale datasets frequently empowers deep models to excel in various downstream fields through transfer learning. Consequently, adopting a compact model developed on ImageNet-1K has become a prevalent strategy when seeking lightweight solutions for RSI classification [14–17]. However, these approaches often entail significant sacrifices in accuracy, as smaller deep models typically possess weaker generalization capabilities. Another common technique involves inserting functional modules into a pre-trained model, which can potentially enhance accuracy with an acceptably increased model size [18–21]. Nevertheless, these methods frequently neglect to re-train the modified model on ImageNet-1K to address the deficiency of sufficient general features in smaller RSI datasets. Consequently, these approaches have not demonstrated significant accuracy improvements due to the diminished advantages of pre-training.

RSIs often contain variable-ground objects that simultaneously belong to multiple categories, a characteristic that distinguishes them from natural images. Therefore, custom-designed deep models with functional modules or layers through human feature engineering could potentially capture the noisy background of RSIs, thereby outperforming ImageNet-1K models [22–30]. This approach could be further optimized by applying the strategy of Neural Architecture Search (NAS) [31–33]. However, all these previous approaches have also bypassed pre-training on ImageNet-1K and have not demonstrated significant advancements in accuracy, even though some of these models have achieved a compact volume.

Knowledge Distillation (KD) [34–35], a technique that transfers knowledge from a robust, complex teacher model to a compact student model, is a promising approach for creating lightweight yet accurate classifiers. KD techniques currently follow two distinct pipelines: transferring knowledge based on intermediate layer features [36–37] or through prediction logits [38–39]. Feature-based approaches often necessitate additional modules within both the teacher and student models to facilitate knowledge transfer [40]. Logit-based methods, on the other hand, typically yield more lightweight student models and incur lower computational costs during the knowledge transfer phase. However, both logit- and feature-based techniques currently lack efficiency, particularly when the teacher model exhibits superior generalization or when there is a significant size disparity between the teacher and student models [41–42]. This issue necessitates a very long training process, potentially involving several tens of thousands of training epochs, to enable the student to match the performance of its teacher.

Among the current literature, there are relatively few approaches that utilize the KD technique to generate lightweight classifiers for RSIs. Furthermore, existing methods often have certain limitations. For example, logit-based methods typically yield poor accuracies [43–45], while feature-based approaches result in large model volumes with average accuracies [46–49]. Moreover, most of these methods have not effectively addressed the efficiency issue in the knowledge transfer process or solved the large accuracy gap between their teacher and student models. The authors believe this problem arises from an oversight of the inherent characteristics of RSIs.

As depicted in Figure 1, the teacher function $f(x)$ that we aim to approximate (represented by the green curve) is dependent on the data distribution of its prediction logit points (indicated in red). When these logit points exhibit significant fluctuations, the student function (shown as the red curve), learned through the KD process, will display a variance of $D(x)$ compared to its teacher. It's important to note that the distribution of logit points is closely tied to the input samples, namely, the training images provided to the models. Consequently, the variance $D(x)$ is intrinsically linked to the inherent characteristics of RSIs.

As illustrated in Figure 2, RSIs exhibit significant intra-class similarity and inter-class differences. For example, the three images from the park, pond, and resort categories

onthe left side of Figure 2 all contain similar water bodies, as highlighted by the red rectangles. Conversely, on the right

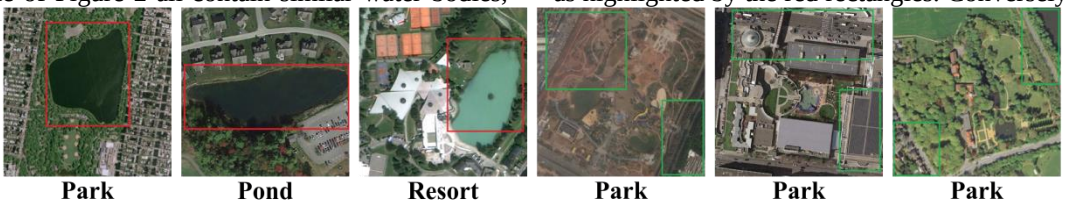

Figure 2. Respective RSI samples among categories with large feature similarity and difference.

side of Figure 2, the three samples from the park category display vastly different backgrounds, as indicated by the green rectangles. This observation suggests that RSIs have a larger data distribution variance compared to natural images. Consequently, we posit that existing logit-based KD techniques, which were developed based on ImageNet-1K, have only sought to minimize the bias between the teacher and student models during the KD process. They have not taken into account the need to reduce the variance in data distribution.

In this paper, we present an efficient and robust knowledge transfer network, termed ERKT-Net, for RSI classification. This work has two unique aspects compared to previous studies. First, we propose a simple strategy to construct a three-CNN ensemble as the teacher model with outstanding accuracy. Second, we introduce a variance-suppression strategy (VSS) and its implementation, the variance-suppression module (VSM), to enhance effectiveness and efficiency during the KD process. Notably, the VSM is a plug-and-play component that only processes the training samples, enabling it to work within any KD pipeline without modifying models.

We evaluated the performance of ERKT-Net on three benchmark RSI datasets, and the results clearly demonstrate its superiority over 40 other methods published between 2020 and 2023. On the challenging NWPU45 dataset, our student model not only achieves a 0.6% improvement in accuracy but also reduces the number of parameters by 88% compared to the top-ranked method in the literature. Additionally, our VSS method reduces training time costs by at least 80% compared to traditional KD approaches. The contributions of this work can be summarized as follows:

(i) We introduce the VSS along with the VSM to improve the efficiency and effectiveness of classical KD techniques for RSI classification. Our VSS method reduces training time costs by at least 80%, with a negligible sacrifice in accuracy.

(ii) We have significantly enhanced the effectiveness of KD techniques in creating compact RSI classifiers. Our lightweight student model achieves a maximum 22.4% increase in accuracy on the challenging NWPU45 dataset compared to other KD methods in the literature.

(iii) Our ERKT-Net is more effective and efficient for developing lightweight yet robust classifiers for RSI tasks. Our KD pipeline is straightforward yet capable of producing more compact and accurate classifiers, even compared to other complex multi-model or feature fusion methods in the literature.

The remainder of this paper is structured as follows: Section 2 reviews the related literature. Section 3 outlines the methodologies, covering the proposed model, the framework, and key settings. Section 4 provides a thorough analysis of the experimental results. Finally, Section 5 presents the study's conclusions.

2. Related works

Using compact models is a common shortcut for developing lightweight RSI classifiers. For instance, Yu *et al.* integrated a MobileNet-V2 model with a shallow bilinear model to create a feature fusion classifier [14]. Chen *et al.* sought to improve a ShuffleNet-V2 model using channel attention [15]. Liang *et al.* combined an EfficientNet-b0 model with recurrent attention modules [16]. Cheng *et al.* proposed a dual-branch model by integrating a CNN and a ViT in parallel [17]. However, the accuracy of these methods remains uncompetitive. In comparison, Alhichri *et al.* [18], Chen *et al.* [19], Zhao *et al.* [20], and Wan *et al.* [21] incorporated attention modules into various pre-trained CNNs, yet there is still significant room for accuracy improvement.

Leveraging human knowledge in feature engineering to develop functional modules or models may yield better solutions than those based on ImageNet-1K models. In this field, Huang *et al.* [22] designed multi-level group convolution modules to enhance a ViT's performance. Xu *et al.* [23] reconstructed CNN features through Lie group structure to develop a lightweight classifier. Wang *et al.* [24] designed a compact CNN with coordinate attention and applied a random depth strategy during training. Additionally, Shi *et al.* [25], Bai *et al.* [26], and Zhang *et al.* [27] each proposed self-compensating convolution, octave convolution, or Laplacian convolution modules as variant CNN classifiers. Bi *et al.* [28] and Guo *et al.* [29] verified multiple granularity feature representation in their CNN classifiers, while Shi *et al.* [30] evaluated their CNN classifier that contains a skip connection at each stage. However, these methods did not re-train their modified models on ImageNet-1K to utilize the advantage of pre-training. However, these methods did not re-train their modified models on ImageNet-1K to utilize the advantages of pre-training. Consequently, most of these methods have not demonstrated significant accuracy improvements compared to fine-tuning pre-trained models [50].

NAS is a promising strategy to find more effective and efficient CNN classifiers for RSIs. For example, Ao *et al.* [31], Broni-Bediako *et al.* [32], and Shen *et al.* [33] have published their NAS work on RSI datasets. However, these NAS approaches are typically conducted on smaller RSI datasets, which only contain tens to hundreds of samples.

We believe that this pipeline cannot leverage the data-driven superiority typically found in large-scale datasets. Consequently, we have not observed competitive accuracy in these NAS works, despite their smaller model sizes.

Hinton *et al.* [35] and other researchers [34] introduced the KD technique to transfer knowledge via prediction logits. However, this KD process often encounters inefficiency, particularly when a teacher model possesses significant generalization capability [41]. In other words, the non-target prediction logits are minuscule with extremely low entropy, causing the information from target logits to significantly suppress that from non-target logits.

As an alternative, Romero *et al.* [36] and other researchers [37, 40] proposed feature-based KD techniques. These techniques involve inserting additional intermediate layers into the teacher and student models to enhance the efficiency of knowledge transfer. Nevertheless, the increased efficiency from feature-based KD techniques comes at a considerably improved cost, particularly in terms of training time and model size.

Furthermore, Zhao *et al.* [38] proposed a decoupled KD loss, divided into target and non-target losses, to mitigate the suppression encountered when using the logit-based KD technique. Concurrently, Huang *et al.* [39] presented another logit-based approach for distilling strong teachers (DIST), employing Pearson distance instead of Kullback-Leibler divergence as the objective function. Despite these advancements, logit-based techniques still require extensive training schemes, often necessitating up to tens of thousands of training epochs to reduce the accuracy gap between a lightweight student and a highly accurate but cumbersome teacher, such as an ensemble of models [42].

In the field of RSI classification, many previous KD approaches have overlooked the issue of information suppression from target logits to non-target ones. Additionally, their accuracy is not highly competitive, and only a few models are of smaller sizes. For example, Chen *et al*. [43] proposed a method for training compact CNN classifiers for RSIs using traditional logit-based KD. Similarly, Xu *et al*. [44] introduced a ViT-teaching-CNN approach through a joint loss for training their teacher and student models concurrently. However, both the teacher and student models within these methods fail to demonstrate exceptional accuracy.

Likewise, Wang *et al.* [45], Li *et al.* [46], Hu *et al.* [47], Xing *et al.* [48], and Zhao *et al.* [49] introduced their CNN or ViT classifiers by utilizing different functional modules for feature-based KD purposes. However, these methods primarily focus on model architecture rather than the efficiency of the KD process. Consequently, they have not exhibited significant accuracy improvements, although some possess significantly increased model sizes.

When employing KD techniques to develop a lightweight yet precise classifier, the importance of a robust teacher model cannot be overstated. Currently, three distinct strategies—namely feature fusion, multiple models, and ensembles of classifiers—prove beneficial for creating precise classifiers. For instance, Zhang *et al.* [51], Lv *et al.* [52], and Wang *et al.* [53] have introduced their ViT-based methods to achieve an RSI classifier with competitive accuracy. Moreover, Li *et al.* [54], Shen *et al.* [55], Tang *et al.* [56], Wang *et al.* [57], and Xu *et al.* [58] have introduced their RSI classifiers, which fuse features from two CNN models. Similarly, Deng *et al.* [59], Zhao *et al.* [60], Ma *et al.* [61], and Wang *et al.* [62] have introduced their RSI classification approaches, which utilize multi-model features of a CNN and a ViT or two ViTs. Additionally, Cheng *et al.* [63] proposed an ensemble of multi-models, where the component classifier is a hidden Markov model refining the features of a CNN. However, most of these methods have only demonstrated remarkable accuracy improvements on small RSI datasets, and their performance significantly degrades when faced with a large RSI dataset or relatively fewer training samples.

To tackle the aforementioned issue, we propose a more effective and efficient KD approach to generate lightweight yet robust RSI classifiers. Our work is primarily rooted in strategies tailored to the inherent nature of RSIs. Initially, we devised a straightforward algorithm to create a stacking ensemble [64], comprising three CNNs with diverse model structures. Subsequently, we incorporate our VSM during the KD process to minimize variances in training samples. Finally, we validate that our method, termed ERKT-Net, surpasses other documented approaches in the literature in terms of effectiveness and efficiency.

3. Methodologies

3.1. Architecture of the Ensemble Teacher

As illustrated in Figure 3 (shown on the next page), the architecture of our ensemble model consists of three distinct CNNs stacked in parallel. Initially, we individually trained the EfficientNet-B0, EfficientNet-B3, and ResNet-50 models using three different training algorithms outlined in our previous works [4, 10, 50], respectively. Subsequently, we utilize three hyperparameters—weight A, weight B, and weight C—to assign weights to the prediction scores of the three CNNs. Finally, we aggregate these weighted scores at a sample-wise dimension to derive the prediction score of the teacher ensemble.

We propose a simple algorithm to configure the three weight parameters within the ensemble. Detailed experiments are presented in Section 4.3, with the results revealing that the accuracies of the ensemble closely match only when EfficientNet-B3 possesses an appropriate value. Therefore, we empirically set the weight of EfficientNet-B3 at 0.4, while the weights of the other models are set to 0.3.

3.2. Method Framework

The framework of the proposed method is depicted in Figure 4. Initially, the VSM processes the original training images, as shown at the figure's apex. Subsequently, the outputs of VSM are simultaneously fed into both the teacher and student models for prediction, as indicated by the green arrows. Thereafter, the prediction scores obtained from the

teacher and student models are employed to compute the KD loss and the student loss, respectively. Ultimately, the parameters of the student model are adjusted via backpropagation, which is illustrated by the red dashed arrows.

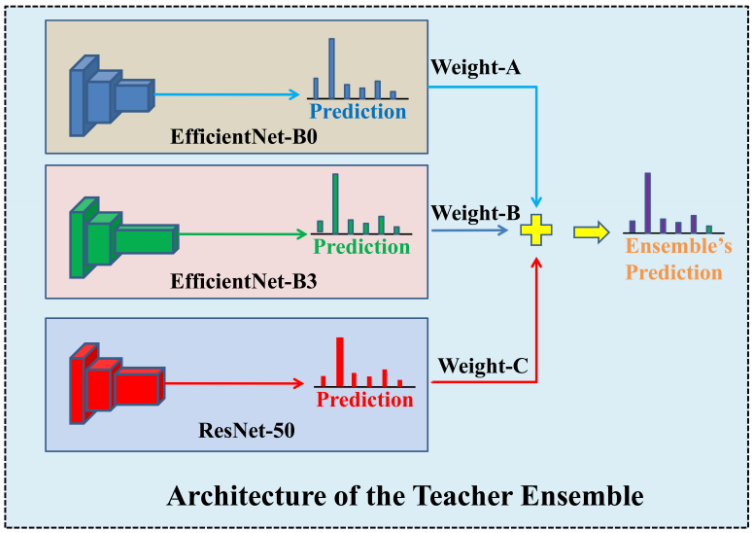

Figure 3. Architectural Overview of the Teacher Ensemble.

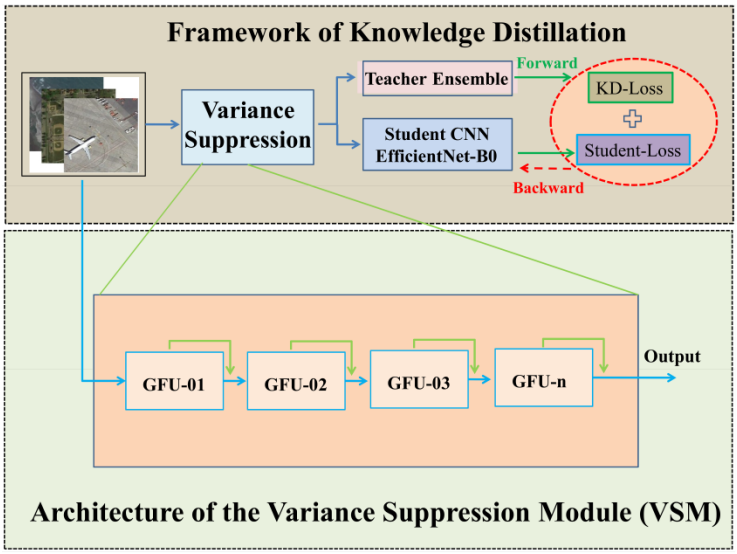

Figure 4. Framework of the Proposed Method.

Table 1. Functions of GFUs within the VSM

DA technique	Operation probability	Parameter settings
Color Jitter	1.0	Brightness, contrast, saturation = 0.5
Random Rotation	1.0	degrees =180
Random Horizontal Flip	0.9	default
Random Vertical Flip	0.9	default
Random Gray scale	0.1	default
Random Auto contrast	0.1	default
Random Erasing	1.0	default
Random Resized Crop	0.1	Size = 176
CutMix	1.0	default

As illustrated at the bottom of Figure 4, the architecture of the VSM comprises several gated functional units (GFUs) in sequence. Specifically, each GFU incorporates a transformation or regularization function to suppress variances in the data distribution of RSIs. Additionally, a shortcut path, indicated by the green fold-line arrows, outputs the unprocessed samples. The purpose of these shortcuts is to achieve appropriate variance suppression.

In each GFU, let x and y represent the input and output images, respectively. The functions within the GFUs are denoted as f_{GFU}. The threshold probability P determines whether a function is activated based on the probability p calculated within the GFU. The workflow in each GFU can be described as follows:

$$y = \begin{cases} f_{GFU}(x), & p \geq P \\ x, & p < P \end{cases}. \quad (1)$$

We have designed nine GFUs within the VSM, and the functions belonging to each GFU are presented in Table 1. These functions are selected from the PyTorch libraries, with the exception of CutMix [65], to facilitate reproducibility for readers. In the 'Operation probability' column of Table 1, a value of 1.0 indicates that the function is always active during training. In the 'Parameter settings' column, 'default' signifies that the parameter settings of the original algorithms remain unaltered.

3.3. Model Architecture

Table 2. Accuracy and Model Sizes of CNN Models.

Model	Accuracy (%)	Params (M)
EfficientNet-B0	77.7	5.3
EfficientNet-B3	82.0	12.2
ResNet-50	76.1	25.6
Teacher Ensemble	None	43.1
Student Model	None	5.3

Our study employs three CNN models to generate the ensemble model: EfficientNet-B0, EfficientNet-B3, and ResNet-50. Additionally, we use EfficientNet-B0 as our student model during the KD process. These CNNs, originally developed for ImageNet-1K, have their detailed structures outlined in reference [3]. A key distinction between EfficientNet and ResNet is the presence of built-in channel attention modules in EfficientNet models. Consequently, EfficientNet outperforms ResNet in terms of accuracy and model size.

Table 2 presents the accuracy (using ImageNet-1K as a test bed) and model sizes of the three CNN models. During the KD phase, we make no structural modifications to the models. Thus, the size of the ensemble model is the combined total of the three CNNs, while the student model retains the size of EfficientNet-B0.

3.4. KD Loss

Consider a RSI dataset, denoted as $S = \{x_i, y_i\}$, where x_i and y_i represent each RSI sample and its corresponding label in S, respectively. In this context, a classifier that accepts x_i as input will produce a prediction logit not only for the target category but also for non-target classes. In contemporary deep learning models, each input x_i is typically normalized to a tensor with values ranging from 0 to 1. Therefore, a classifier can essentially be viewed as a function, denoted as f, which accepts tensors as input and outputs logit vectors. Assuming that the number of categories in S is represented as c, then the function f can be described as follows:

$$y_{i,c} = f(x_i). \quad (2)$$

A decade ago, classifiers that utilized human feature engineering were typically shallow models. These models were compact but had limited generalization capabilities. In contrast, current deep learning models, which employ automatic feature extraction, exhibit superior generalization capabilities but are often large in size. Theoretically, in the simplest case, a deep model with more learnable parameters will typically fit a dataset better than a model with fewer parameters. As a result, a deep model with high accuracy is often cumbersome, particularly for challenging tasks. In this context, deployment becomes challenging when the requirements for model inference speed are stringent or when hardware resources are limited.

Bucila *et al.* [34] introduced the concept of model compression, which involves using a compact (student) model to approximate a more robust but larger (teacher) model. Building on this, Hinton *et al.* [35] further developed this knowledge transfer technique and coined it knowledge distillation (KD).

If we denote the output logits of a model as z_i, the softmax function will transform z_i into probabilities corresponding to each category, denoted as p_i. This transformation can be described as follows:

$$p_i = softmax(z_i) = \frac{\exp(z_i)}{\sum_{i=1}^{c} \exp(z_i)}. \quad (3)$$

Logit-based KD techniques typically employ the Kullback-Leibler (KL) divergence as the objective function. During training, this function takes the probabilities from both the teacher and student models as inputs and calculates the losses. The loss, denoted as $\mathcal{L}_{KD}$, can be expressed as follows:

$$\mathcal{L}_{KD} = KL(P_t \parallel P_s) = \sum_{i=1}^{c}(P_{t,i}\ log\frac{P_{t,i}}{P_{s,i}}), \quad (4)$$

where $P_{t,i}$ and $P_{s,i}$ are the probabilities from the teacher and student models, respectively.

In a robust model, the prediction probability for target categories typically reaches up to 98%, while for non-target classes, it may be as low as 0.1% or even less. Consequently, the logits for target categories tend to suppress those for non-target ones, especially when the teacher model exhibits high accuracy. To address this issue, Hinton *et al.* introduced a hyperparameter known as temperature, denoted as t, to soften the model's prediction logits. Specifically, they incorporated t into equation (4), thereby altering the distribution of the logit data. The softened loss, denoted as $\mathcal{L}_{\mathrm{KD}}^{t}$, can be described as follows:

$$p_i^t = softmax\left(\frac{z_i}{t}\right). \quad (5)$$

$$\mathcal{L}_{\mathrm{KD}}^{t} = \sum_{i=1}^{c}(t^2 \times P_t^t\ log\frac{P_t^t}{P_s^t}). \quad (6)$$

Moreover, integrating the cross-entropy loss of the student model with $\mathcal{L}_{\mathrm{KD}}^{t}$ can expedite convergence during the KD process. Thus, the standard KD loss, denoted as $\mathcal{L}_{\mathrm{training}}$, can be expressed as follows:

$$\mathcal{L}_{\mathrm{training}} = -\sum_{\mathrm{i=1}}^{\mathrm{c}}(y_i \log P_s) + \sum_{i=1}^{c}(t^2 \times P_t^t\ log\frac{P_t^t}{P_s^t}). \quad (7)$$

However, Stanton *et al.* [41] observed that when dealing with a robust teacher model or a significant accuracy gap between teacher and student models, the student struggles to match the teacher through the standard KD process. Similarly, Beyer *et al.* [42] confirmed that resolving the large accuracy discrepancy between the teacher and student models in the standard KD process requires an extremely high number of training epochs, potentially up to tens of thousands.

Recently, Zhao *et al.* [38] introduced a variant of the standard KD loss, termed decoupled KD loss, to enhance the efficiency of the KD process. This loss includes two components: target and non-target losses. The method replaces non-target logits with minor values when calculating target loss, and vice versa.

Huang *et al.* [39] introduced an alternative loss function, termed DIST, which employs the Pearson distance in place of the conventional KD loss defined in equation (6). The Pearson correlation coefficient and the Pearson distance, denoted as ρ and D_P, respectively, are defined as follows:

$$D_P = 1 - \rho(V_t, V_s) = 1 - \frac{\sum_{i=1}^{c}(V_t - \overline{V_t})(V_s - \overline{V_s})}{\sqrt{\sum_{i=1}^{c}(V_t - \overline{V_t})^2 \sum_{i=1}^{c}(V_s - \overline{V_s})^2}}, \quad (8)$$

where V_t and V_s represent the prediction vectors of the teacher and student models, respectively. To enhance the information entropy during the distillation phase, DIST defines the result as the inter-class loss, denoted as $\mathcal{L}_{inter}$, which is calculated using equation (8) with V_t and V_s as inputs. If we denote the training batch size as N, $\mathcal{L}_{inter}$ can be expressed as:

$$\mathcal{L}_{inter} = \frac{1}{N}\sum_{i=1}^{N} D_P(V_t, V_s). \quad (9)$$

Moreover, DIST introduces another intra-class loss, denoted as $\mathcal{L}_{intra}$, which is computed using equation (8) with the N and M dimension transposes of V_t and V_s as inputs. Here, M represents the number of categories in a dataset. Therefore, $\mathcal{L}_{intra}$ can be expressed as follows:

$$\mathcal{L}_{intra} = \frac{1}{M}\sum_{j=1}^{M} D_P(V_t^T, V_s^T). \quad (10)$$

While both the decoupled KD and DIST techniques have demonstrated substantial progress in narrowing the accuracy gap between teacher and student models, neither method effectively addresses the variations in the data distribution of training samples, a fundamental characteristic of RSIs. We found that simply applying these KD techniques, initially developed for natural images, may not be optimal for many RSI tasks. This is particularly true when the teacher model is a robust ensemble with a significant volume gap compared to the student. Consequently, we introduce our ERKT-Net as a novel KD method for RSI classification, specifically designed to accommodate the inherent characteristics of RSIs. The loss function of ERKT-Net is presented as follows:

Initially, we retained the cross-entropy objective function but employed a hyperparameter α to adjust the loss value when CutMix is activated. The cross-entropy loss, denoted as L_{cross}, the loss of a class-A sample as L_A, and the loss of another class-B sample as as L_B, are defined. Consequently, L_{cross} when CutMix is active can be expressed as:

$$L_{cross} = (1 - \alpha) \times L_A + \alpha \times L_B. \quad (11)$$

Here, α represents the ratio of the class-B patch size to the class-A image. α is randomly sampled from the beta distribution of (0, 1).

Subsequently, we selected DIST as our KD loss and assigned the same weight of 2.0 to the inter- and intra-losses, as Huang *et al.* [39] suggested. Additionally, we set the temperature hyperparameter at 2.0 without further optimization. Therefore, the training loss of our ERKT-Net, denoted as $Loss_{ERKT-Net}$, can be formulated as:

$$Loss_{ERKT-Net} = L_{cross} + 2 \times (\mathcal{L}_{inter} + \mathcal{L}_{intra}). \quad (12)$$

3.5. Training Algorithm

The training procedures for the distillation phase are outlined in Algorithm 1 (shown on the next page), represented in pseudo-code.

As indicated in line 1, the entire distillation process spans 600 training epochs. We empirically set this threshold at 600 epochs as a benchmark because our ERKT-Net can more efficiently transfer knowledge from the teacher model compared to the tens of thousands of epochs required by the classic KD technique. Lines 2 and 3 demonstrate that a batch of 30 images and their corresponding labels are first processed by the VSM. The outputs from the VSM are then fed into the teacher and student models simultaneously. Following this, as depicted in lines 4, 5, 6, and 7, the prediction logits of both the teacher and student are input into the loss function. Subsequently, gradients are calculated to update the parameters of the student model. Finally, as shown in lines 9 and 11, the accuracy of the student model is verified at the end of each epoch, and a record of the accuracy at each epoch is reported upon the completion of the training. In summary, the uniqueness of Algorithm 1 lies in the involvement of the VSM in the KD process, enhancing the effectiveness and efficiency of our ERKT-Net.

Regarding the other hyperparameter settings in training, we established the initial learning rate at 2E-04, which is managed using the cosine decay algorithm. The Adam-W optimizer was employed, with a weight decay set at 1E-06. Concurrently, a fixed resolution of 256^2 was maintained during both the training and testing stages for all datasets.

Algorithm 1. Training procedure using pseudo-code

Definitions: The training subset for RSI is denoted as $S_{train} = \{(x_i, y_i)\}$, while the testing set is denoted as $S_{test} = \{(x_i, y_i)\}$. The ensemble-teacher model is represented by f_t; and the EfficientNet-b0 student model is represented by f_s. The VSM is signified by f_{VSM}; The CutMix algorithm is denoted as f_{CM}, and the distillation loss function is represented by $\mathcal{L}_{ERKT-Net}$.
Input: Images and labels from training or testing subsets.
Output: Student classifier's accuracy (Acc) results.

For Epoch = 1, 2, . . . , 600 **Do**
  For iteration = 1 to $\left(\frac{length(S_{train})}{30} + 1\right)$ **Do**
    Sample a batch of samples from S_{train},
    and input them to the functions f_t and f_s, respectively.
    Predict teacher probabilities using the equation: $y_i^t = f_t(f_{CM}(f_{VSM}(x_i)))$.
    Predict student probabilities using the equation: $y_i^s = f_s(f_{CM}((f_{VSM}(x_i))))$.
    Calculate the loss using the equation: $Loss = \mathcal{L}_{ERKT-Net}(y_i^t, y_i^s)$.
    Update parameters through back propagation.
  End For
  Calculate the student model's accuracy using the equation:
  $Acc = (f_s(x_i) == y_i)$, where $x_i, y_i \in S_{test}$, and save the Acc result.
End For
Return the Acc results

Figure 5. Exemplary samples from each category in the AID30 dataset.

Figure 6. Exemplary samples from each category in the NWPU45 dataset

Figure 7. Exemplary samples from each category in the AFGR50 dataset

3. 6. Dataset and Division

We utilized three RSI datasets to compare the performance of our method. The first two, the AID30 and the NWPU45 dataset, are widely recognized benchmarks in existing literature. Additionally, we employed the AFGR50 dataset to validate the effectiveness of our method.The AID30 dataset consists of 30 categories, with a total of 10,000 images, all of which have a uniform resolution of 600^2. The NWPU45 dataset comprises 45 categories, with a total of 31,500 images, all maintaining a consistentresolution of 256^2. The AFGR50 dataset contains 50 categories, with a total of 12,500 images, all sharing a resolution of 128^2.

Both the NWPU45 and AFGR50 datasets are balanced, with each category containing 700 and 250 images, respectively. However, the AID30 dataset is imbalanced, with a varying number of 220 to 420 samples per class. Figures 5, 6, and 7 showcase representative samples from each category for the three datasets.

To ensure a fair comparison, we adhered to the same training ratio (TR) as outlined in the literature: 20% and 50% for AID30; 10% and 20% for NWPU45; and 10%, 20%, and 30% for AFGR50. For each TR, we randomly selected samples from the entire dataset to form the training subsets, with the remaining samples designated as testing subsets.

3.8. Performance Evaluation Metrics

In our research, we utilized overall accuracy (OA) and the confusion matrix as performance evaluation metrics, as they are prevalent in the current literature and offer abundant experimental results for comparing classifications in RSI studies.

OA is a metric that quantifies the proportion of correct predictions made by a model out of the total number of predictions. High accuracy reflects a model's robustness and its ability to generalize from the training data to unseen data. The symbol N_c denotes the total number of samples correctly classified, while N_t signifies the total number of classified samples. Therefore, OA can be expressed as:

$$OA = \frac{N_c}{N_t}. \quad (13)$$

The confusion matrix is a tabular visualization that enables the assessment of a classification model's performance. It contrasts the actual target values with those predicted by the model, providing insight into the types of errors made. The matrix is divided into four quadrants: true positives, true negatives, false positives, and false negatives. True positives and negatives correspond to correct predictions, while false positives and negatives represent errors. This matrix is particularly valuable in elucidating the model's predictive capabilities across different classes, which is essential for tasks with imbalanced datasets or when the costs of different types of errors vary significantly.

3.9. Experimental Environments

Experiments were conducted on four computers, each with a Nvidia 2060 Graphics Processing Unit (GPU), using PyTorch version 1.11.0 on Windows 10. The reported results, derived from at least three runs, represent either mean values or deviations to ensure reliability.

4. Experimental Results

4.1. OA Results

Tables 3, 4, and 5 display the OA comparison results for the AID30, NWPU45, and AFGR50 datasets, encompassing a total of 42 methods. The 'Parameters' column in these tables contains either original data or evaluations derived from the model backbones, as outlined in the relevant literature. The term 'None' within the tables signifies the absence of detailed information in the associated literature.

Table 3: A Comparative Study on OAs of Different Methods Using the AID30 Dataset.

Models	Method Uniqueness	Params (M)	AID (%)	
			TR-20%	TR-50%
BiMobileNet [14]	Compact Model	7.8	94.83 ± 0.24	96.87 ± 0.23
RSC-Net [15]		1.3	None	96.24
ERA-Net [16]		6.7	95.93 ± 0.13	98.39 ± 0.16
LDBST-Net [17]		9.3	95.10 ± 0.09	96.84 ± 0.20
EfficientNet-B3-Attn-2 [18]	Inserting Attention Modules	>12.0	94.45 ± 0.76	96.56 ± 0.12
MBLANet [19]		>25.6	95.60 ± 0.17	97.14 ± 0.03
EAM-Net [20]		>46.8	94.26 ± 0.11	97.06 ± 0.19
LmNet [21]		>25.0	95.82 ± 0.25	97.12 ± 0.14
LTNet [22]	Custom Designed Model	8.2	94.98 ± 0.08	None
LGRINet [23]		4.6	94.74 ± 0.23	97.65 ± 0.25
LRSCM-Net [24]		7.6	95.41	97.28
SC-CNN [25]		0.5	93.15 ± 0.25	97.31 ± 0.10
MF2CNet [26]		33.2	95.54 ± 0.17	97.02 ± 0.28
LHNet [27]		>46.8	93.30 ± 0.10	97.81 ± 0.13
AGOS-Net [28]		>12.5	95.81 ± 0.25	97.43 ± 0.21
SEINet [30]		2.9	95.37 ± 0.09	**98.61** ± 0.16
SLGE-Net [32]	NAS	5.1	None	96.10 ± 0.18
AF-NAS-Net [33]		3.8	95.65 (TR-60%)	
TST-Net [43]	Logits-based KD	1.0	85.50	None
ET-GSNet [44]		11.7	95.58 ± 0.18	96.88 ± 0.19
LaST-Net [45]	Features-based KD	28	83.23	87.34
DKA-Net [46]		4.4	95.09	96.94
VSDNet [47]		>8.0	96.73 ± 0.15	97.95 ± 0.10
ESD-MBENet [49]		23.9	96.39 ± 0.21	**98.40** ± 0.23
TRS-Net [51]	Single ViT	46.3	95.54 ± 0.18	**98.48** ± 0.06
SC-ViT [52]		40.1	95.56±0.17	96.98±0.16
ViT-AEv2 [53]		18.8	96.91 ± 0.06	98.22 ± 0.09
GRMA-Net [54]	Multiple CNNs	54.1	96.19 ± 0.48	97.84 ± 0.39
ACGLNet [55]		33.6	94.44 ± 0.09	96.10 ± 0.10
ACNet [56]		>276.6	93.33 ± 0.29	95.38 ± 0.29
T-CNN [57]		15.9	94.55 ± 0.27	96.72 ± 0.23
GLDBS-Net [58]		>23.4	95.45 ± 0.19	97.01 ± 0.22
CTNet [59]	Multiple models	>107.8	96.25 ± 0.10	97.70 ± 0.11
L2RCF-Net [60]		46.7	**97.00** ± 0.17	97.80 ± 0.22
HHTL-Net [61]		None	96.52 ± 0.13	96.88 ± 0.21
CNN–HMM [63]	CNN Ensemble	19	93.93 ± 0.15	97.81 ± 0.04
CNN Ensemble Teacher	Our KD	43.1	97.36 ± 0.14	98.31 ± 0.18
ERKT-Net (this work)		5.3	**97.20** ± 0.08	**98.19**± 0.13

4.1.1. OA results for AID30.

As demonstrated in Table 3, our ERKT-Net (the student model) exhibits a minor OA discrepancy of approximately 0.1% compared to the ensemble teacher at both 20% and 50% TRs. This outcome suggests that despite the significant model volume disparity between the teacher and student, the knowledge transfer in our KD process is effective and robust. When compared to other KD methods, ERKT-Net outperforms all, with OA improvement values ranging from 0.5% to 11.7% at the 20% TR. However, the OA improvements diminish as the number of training samples increases to 50% TR. This finding further substantiates that ERKT-Net is a more effective KD technique when the objective is to achieve a lightweight yet accurate classifier.

Based on the 20% TR results, ERKT-Net surpasses all other methods, including those utilizing multiple models with 10 to 20 times the volume. This outcome suggests that ERKT-Net possesses superior generalization capability even when the number of training samples is limited. However, based on the 50% TR results, three methods, namely SEINet [30], ESD-MBENet [49], and TRS-Net [51], exhibit approximately 0.2% to 0.4% higher OA values (as highlighted in red). This discrepancy may be attributed to three factors.

Firstly, AID30 is an unbalanced dataset with an average of 333 samples per category. However, some of the most confusing classes of AID30, such as center, church, and resort, only include 260, 240, and 290 samples, respectively. The data distribution generated by a randomly selected subset may cause fluctuations in the accuracy of classifiers. Secondly, ESD-MBENet and TRS-Net have a larger capacity. A larger model has the advantage of retaining more features in a dataset as the number of training samples increases. Thirdly, our teacher ensemble is suboptimal because it was generated using an empirical stacking

Table 4: A Comparative Study on OAs of Different Methods Using the NWPU45 Dataset

Models	Method Uniqueness	Params (M)	NWPU (%)	
			TR-10%	TR-20%
BiMobileNet [14]	Compact Model	7.8	92.06 ± 0.14	94.08 ± 0.11
RSC-Net [15]		4.6	91.91 ± 0.15	94.43 ± 0.16
ERANet [16]		6.7	91.95 ± 0.19	95.12 ± 0.17
LDBST-Net [17]		9.3	93.86±0.18	94.36±0.12
MBLANet [19]	Inserting Attention Modules	>25.6	92.32 ± 0.15	94.66 ± 0.11
EAM-Net [20]		>46.8	91.91 ± 0.22	94.29 ± 0.09
LmNet [21]		>25.0	93.00 ± 0.11	94.85 ± 0.14
LTNet [22]	Custom Designed Model	8.2	92.21 ± 0.11	None
LGRINet [23]		4.6	91.95 ± 0.15	94.43 ± 0.16
LRSCM-Net [24]		7.6	92.18	94.74
SC-CNN [25]		0.5	92.02 ± 0.50	94.39 ± 0.16
MF2CNet [26]		33.2	92.07 ± 0.22	93.85 ± 0.27
LHNet [27]		>46.8	89.89 ± 0.15	92.53 ± 0.13
AGOS-Net [28]		>12.5	93.04 ± 0.35	94.91 ± 0.17
MGS-Net [29]		244.2	91.92 ± 0.12	94.33 ± 0.08
SEINet [30]		2.9	92.98 ± 0.11	95.35 ± 0.16
TPENAS-Net [31]	NAS	1.7	None	90.38
SLGE-Net [32]		5.1	None	96.56 ± 0.13
AF-NAS-Net [33]		3.8	95.32 (TR-60%)	
TST-Net (Chen *et al.*, 2018)	Logits-based KD	1.0	80.00(TR-50%)	
ET-GSNet [44]		11.7	92.72 ± 0.28	94.50 ± 0.18
LaST-Net [45]	Feature-based KD	28	72.58	73.67
DKA-Net [46]		4.4	93.72	95.76
VSDNet [47]		>8.0	93.24 ± 0.11	95.67 ± 0.11
CKD-Net [48]		None	0.916 (TR is not clear)	
ESD-MBENet [49]		23.9	93.05 ± 0.18	95.36 ± 0.14
TRS-Net [51]	Single ViT	46.3	93.06 ± 0.11	95.56 ± 0.20
SC-ViT [52]		40.1	92.72±0.04	94.66±0.10
ViT-AEv2 [53]		18.8	94.41 ± 0.11	95.60 ± 0.06
GRMA-Net [54]	Multiple CNNs	54.1	93.67 ± 0.21	95.32 ± 0.28
ACNet [56]		>276.6	91.09 ± 0.13	92.42 ± 0.16
T-CNN [57]		15.9	90.25 ± 0.14	93.05 ± 0.12
GLDBS-Net [58]		>23.4	92.24 ± 0.21	94.46 ± 0.15
CTNet [59]	Multiple models	>107.8	93.90 ± 0.14	95.40 ± 0.15
L2RCF-Net [60]		46.7	94.58 ± 0.16	95.60 ± 0.12
HHTL-Net [61]		None	92.07 ± 0.44	94.21 ± 0.09
P2FEViT [62]		>112.2	**94.97** ± 0.13	**95.74** ± 0.19
CNN–HMM [63]	CNN ensemble	19	93.43 ± 0.25	95.51 ± 0.21
CNN Ensemble Teacher	Our KD	43.1	95.11 ± 0.06	96.62 ± 0.06
ERKT-Net (this work)		5.3	**94.90** ± 0.05	**96.36**± 0.05

strategy without an optimization search for the weight hyperparameters.

Therefore, using the 20% TR of AID30 as a test bed is more objective.

4.1.2. OA results for NWPU45.

Table 4 demonstrates that the accuracy discrepancy between our ERKT-Net and the teacher remains minimal at the challenging NWPU45, standing at approximately 0.2%. This result suggests that our KD process maintains its effectiveness and robustness, even though the NWPU45 dataset is three times larger and contains more confusing categories. In comparison to other KD methods, ERKT-Net stands out, with OA improvement values spanning from 1.2% to 22.3% at the 10% TR. Furthermore, the OA improvements persist at the 20% TR, with amplified values ranging from 0.7% to 22.4%. This evidence corroborates that ERKT-Net is more effective when the objective is to utilize KD techniques to construct a lightweight yet precise classifier for a challenging RSI dataset.

According to the results obtained with 10% TR, ERKT-Net outperforms all other methods, except for P2FEViT [62]. P2FEViT shows comparable accuracy, but its model size is 8.4 times larger. However, when the TR is increased to 20%, ERKT-Net surpasses all other methods, achieving an OA that is at least 0.7% higher than P2FEViT. This observation underscores the consistent superior generalization capability of ERKT-Net, even when evaluated against the highly challenging NWPU45 dataset.

When comparing the results between the AID30 and NWPU45, it is observed that methods with a slight accuracy advantage at the 50% TR of AID30, such as SEINet, ESD-

Table 5: A Comparative Study on OAs of Different Methods Using the AFGR50 Dataset

Models	Method Uniqueness	Params (M)	AFGR50 (%)		
			TR-10%	TR-20%	TR-30%
P2FEViT [62]	Multiple Models	>112.2	89.24 ± 0.10	95.22 ± 0.13	**97.27** ± 0.15
ResNet-50 [50]	Single CNN	25.6	90.29 ± 0.11	94.93 ± 0.32	96.67 ± 0.02
EfficientNet-b0 [10]	Single CNN	5.3	90.21 ± 0.48	95.46 ± 0.27	96.82 ± 0.25
CNN Ensemble Teacher	Our KD	43.0	92.44 ± 0.50	96.57 ± 0.21	97.41 ± 0.26
ERKT-Net (proposed)		5.3	**92.02** ± 0.24	**96.37** ± 0.16	**97.26** ± 0.19

MBENet, and TRS-Net, exhibit significant degradation in their generalization capability when applied to a challenging RSI dataset. Specifically, these methods all presented an OA gap value of 1% to 2% less than our ERKT-Net. This observation confirms that the superior performance of SEINet, ESD-MBENet, and TRS-Net is highly dependent on the ample training samples from the unbalanced AID30 dataset.

4.1.3. OA results for AFGR50

The AFGR50, a novel fine-grained RSI dataset, was introduced in March 2023. It comprises 12,500 aircraft images, each with a fixed resolution of 128^2. The dataset is organized into 50 categories, with each category containing 250 samples. To date, comparable studies on this dataset are limited, with P2FEViT being a notable exception. Given that fine-grained image recognition is a common task in RSI, we assert that AFGR50 is an appropriate benchmark for evaluating the generalization capability of our proposed ERKT-Net on fine-grained RSIs.

Table 5 presents the OA results for four methods on the AFGR50 dataset. To ensure a more objective comparison, we have also included the results of two distinct single-CNN methods, each trained using a different algorithm.

As demonstrated in Table 5, ERKT-Net surpasses other models in terms of OA, with the exception of P2FEViT at 30% TR, which equals ERKT-Net. This is primarily because P2FEViT benefits from a scaling effect advantage due to its larger capacity. These results highlight the robustness of our ERKT-Net in recognizing fine-grained RSIs. Moreover, as shown in line 3, ERKT-Net, which uses the same EfficientNet-b0 model but with a different learning approach, exhibits superior OA. This suggests that ERKT-Net has effectively learned from the knowledge contained within the teacher ensemble.

4.1.4. Overview of OA Comparisons

The consistency of the OA disparity results between ERKT-Net and the teacher ensemble, as shown in Tables 3, 4, and 5, attests to ERKT-Net's ability to effectively assimilate the majority of the 'dark knowledge' contained within the teacher, with only a minor compromise in accuracy. When compared to other KD methods, ERKT-Net's superior performance demonstrates its efficiency and robustness in knowledge transfer.

In a similar vein, when comparing other strategies aimed at developing a more lightweight or accurate classifier for RSI recognition, the OA results presented in Tables 3, 4, and 5 suggest that ERKT-Net is a more effective approach for achieving a lightweight yet accurate classifier.

4.2. Confusion Matrixes

To analyze the confusion results for each category, we present the confusion matrices in Figures 8 and 9 for AID30 and NWPU45 (shown on the next page), respectively, both at the same TR-20%. In these figures, an OA of 100% corresponds to 1.0. The OA values highlighted in red indicate the most confusing categories, while the OA values in blue represent categories with a larger ratio of misclassified samples. For ease of interpretation, we have marked the most frequently confused categories with a red rectangle.

4.2.1. Examination of Confusion Results for AID30.

As Figure 8 illustrates, the challenging categories within AID30 are the center, park, resort, school, and square, all of which exhibit OA values below 91%. Additionally, two other classes, the church and industry, display OA values below 97%. Conversely, all remaining categories show OA values of 97% or higher. These findings indicate that only seven categories within AID30 are particularly challenging, especially those with fewer samples than average.

When comparing these results with methods that demonstrate high accuracy at 50% TR, such as ESD-MBENet [49], it is observed that the challenging categories remain the same. However, ESD-MBENet exhibits lower OA values than ERKT-Net in other categories, including the railway station and bare land. This discrepancy suggests that ESD-MBENet's performance is more dependent on a large number of training samples than an enhanced recognition capability for specific classes.

4.2.2. Examination of Confusion Results for NWPU45.

To enhance overall readability, we removed the categories with OA above 98.0%. As depicted in Figure 9, the challenging categories in NWPU45 are the church, lake, palace, and railway station, all exhibiting OA values below 92.1%. Additionally, twelve other classes display OA values below 96%. These findings indicate that sixteen categories, or one-third of the total in NWPU45, are confusing. Compared to AID30, these confusing categories in NWPU45 include both artificial scenes and natural fields. Therefore, using NWPU45 as an evaluation benchmark for a model's generalization capability is more challenging yet objective.

When contrasting these results with other methods in the literature, no significant disparities in confusing categories emerge. Furthermore, the methods that excelled under the

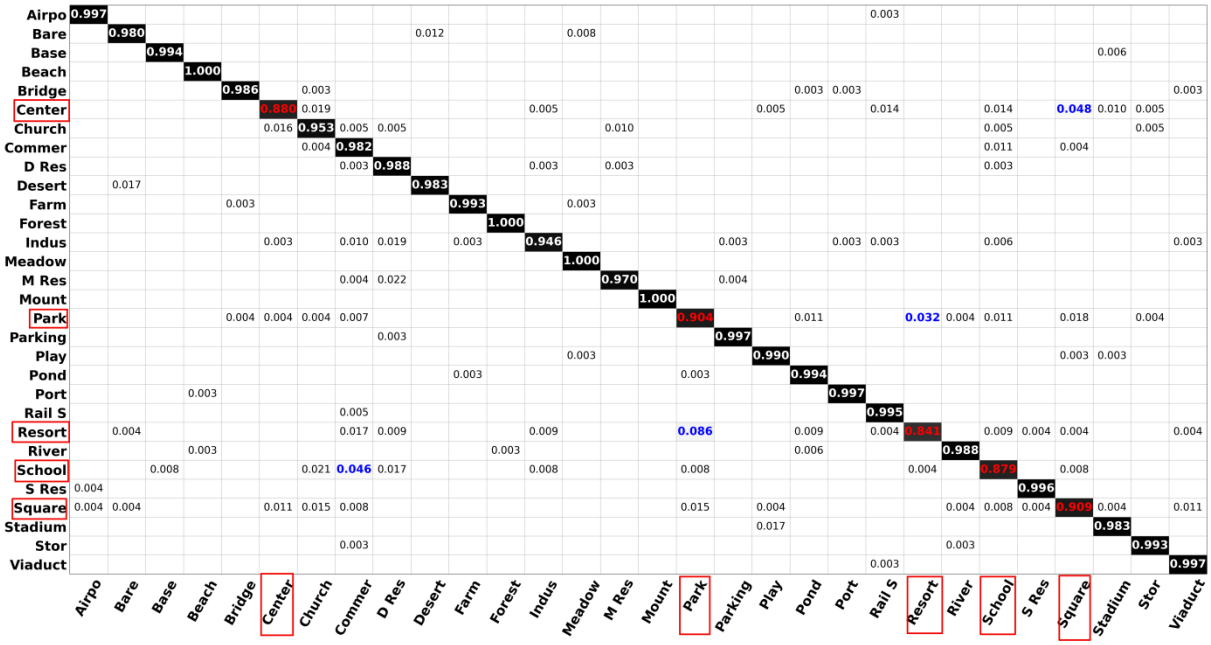

Figure 8. Confusion Matrix for AID30 with a TR of 20%

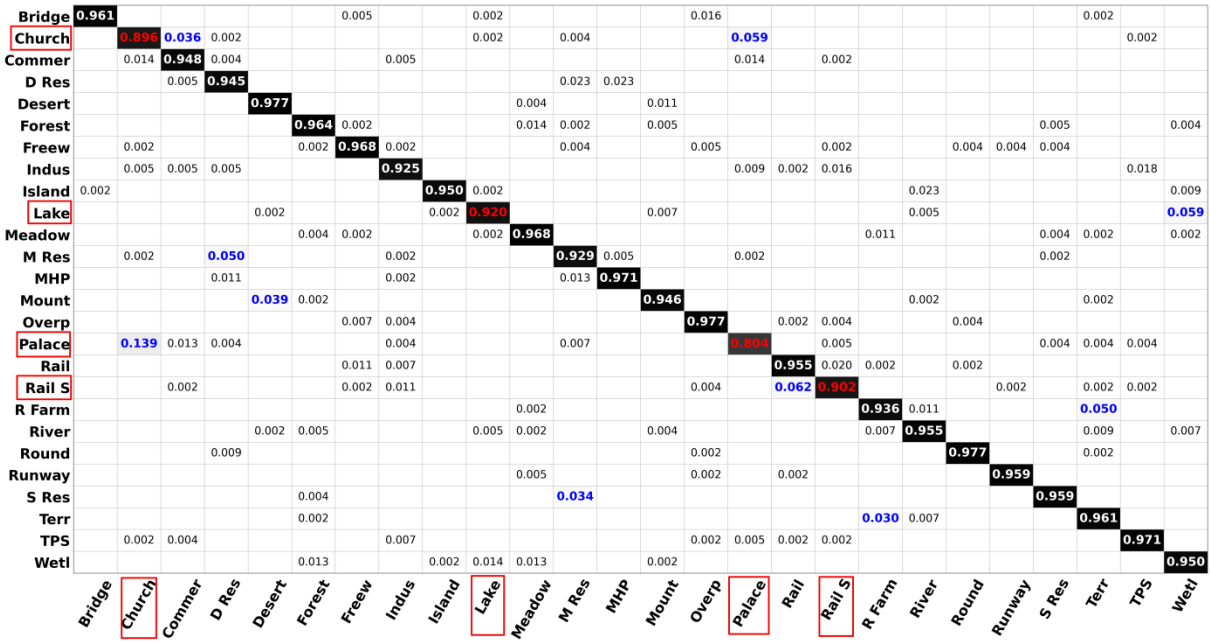

Figure 9. Confusion Matrix for NWPU45 with a TR of 20%.

50% TR on AID30 exhibited inferior performance on NWPU45. This consistent trend highlights the imbalanced nature of AID30, resulting in fluctuations in accuracy.

4.3. Sensitivity Analysis

Two settings may significantly impact the performance of our ERKT-Net. The first setting concerns the three weight parameters within the teacher ensemble, while the second relates to the ratio between the cross-entropy loss and the DIST loss defined in equation (12). We conducted extensive experiments to validate that these two settings have a controllable influence on our method.

Initially, we proposed a straightforward algorithm to generate our teacher ensemble. Let W_{B3}, W_{B0}, and W_{R50} denote the weights for EfficientNet-B3, EfficientNet-B0, and ResNet-50, respectively. The relationship between these three weights can be expressed as follows:

$$W_{B0} = W_{R50} = (1 - W_{B3}) \times 0.5. \quad (14)$$

Subsequently, we increased the weight for EfficientNet-B3 from 0.1 to 0.9 at intervals of 0.1. Thus, we obtained nine candidate ensemble models and verified their OAs on the AID30 and NWPU45 datasets. As shown in Table 6, the results reveal that the OAs of the candidate ensembles on the AID30 or NWPU45 datasets closely match when the

Table 6: OA of Candidate Ensemble Models on AID30 and NWPU45

Weight for EfficientNet-B3	ERKT-Net's OA (%)	
	AID30 (TR-20%)	NWPU45 (TR-10%)
0.1	97.14 ± 0.07	95.02 ± 0.04
0.2	97.23 ± 0.02	95.09 ± 0.09
0.3	97.33 ± 0.07	**95.14** ± 0.07
0.4	**97.36** ± 0.14	95.11 ± 0.06
0.5	97.29 ± 0.15	95.06 ± 0.07
0.6	97.28 ± 0.14	94.94 ± 0.05
0.7	97.19 ± 0.14	94.83 ± 0.08
0.8	97.11 ± 0.11	94.72 ± 0.09
0.9	97.00 ± 0.09	94.61 ± 0.10

Table 7: Performance of ERKT-Net on AID30 and NWPU45 with Varying Loss Ratios

Ratio of two Losses	ERKT-Net's OA (%)	
	AID30 (TR-20%)	NWPU45 (TR-10%)
0.0	97.13 ± 0.02	94.76 ± 0.15
0.5	97.16 ± 0.05	94.88 ± 0.08
1.0	**97.20** ± 0.08	**94.90** ± 0.05
1.5	97.10 ± 0.05	94.83 ± 0.09
2.0	96.81 ± 0.06	94.28 ± 0.07

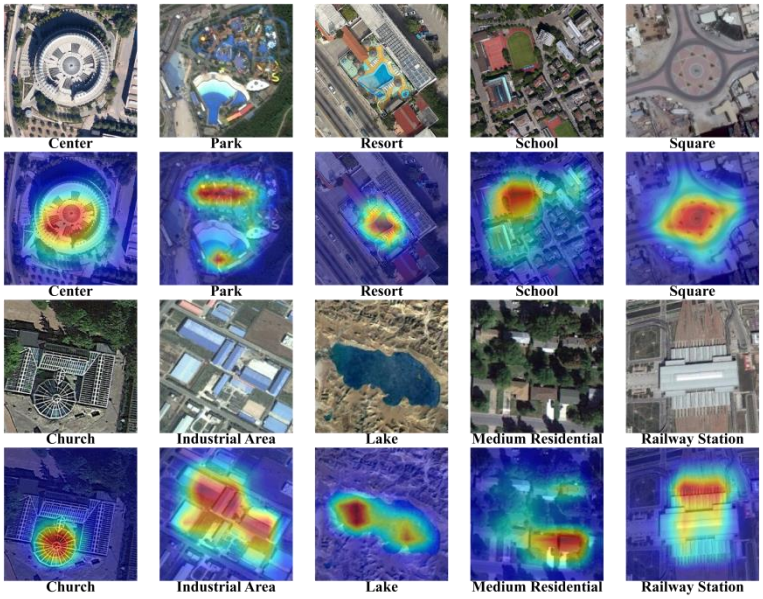

Figure 10. Grad-CAM Analysis for Representative RSI Samples.

weight for EfficientNet-B3 is around 0.4. Therefore, we empirically set the weight for EfficientNet-B3 at 0.4 and the others at 0.3, as the accuracy of the ensemble is not highly sensitive across different datasets.

Furthermore, we validate the influence of the ratio between the two losses. Let R denote the ratio, and let $\mathcal{L}_{\text{DIST}}$ denote $2 \times (\mathcal{L}_{inter} + \mathcal{L}_{intra})$ as defined in equation (12). Equation (12) can be reformulated as follows:

$$Loss_{ERKT-Net} = R \times L_{cross} + (2 - \text{R}) \times \mathcal{L}_{\text{DIST}}. \quad (15)$$

We increment R from 0.0 to 2.0 at intervals of 0.5 and then evaluate its effectiveness. As depicted in Table 7, our ERKT-Net demonstrates improved performance on the AID30 and NWPU45 datasets when R equals 1.0. Therefore, we employ this result as our method's final setting.

4.4. Visualization and Analysis

In this section, we utilize two techniques to illustrate the activation and feature effectiveness of ERKT-Net. First, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [66] to provide visual explanations for the model's predictions. Second, we use t-Distributed Stochastic Neighbor Embedding [67], commonly referred to as t-SNE, to analyze the effectiveness of the model's features.

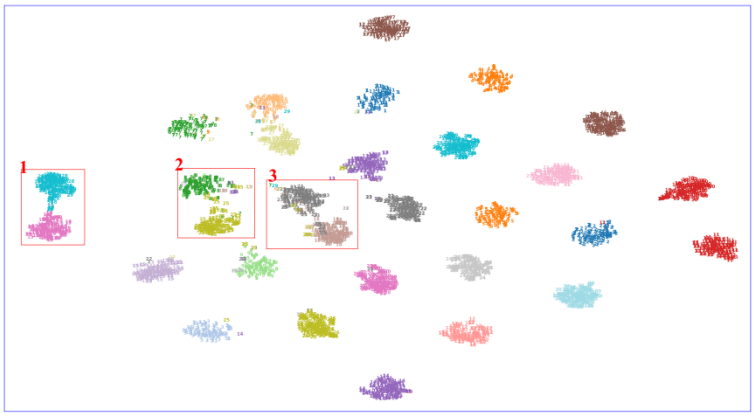

Figure 11. t-SNE Visualization on the AID30 Dataset

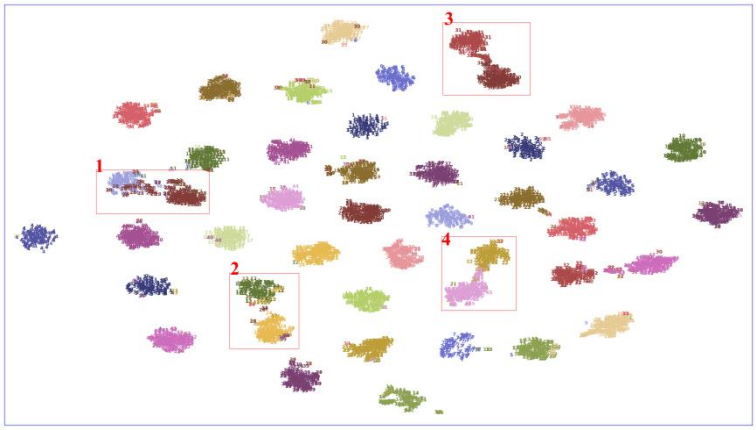

Figure 12. t-SNE Visualization on the NWPU45 Dataset

4.4.1. Grad-CAM results.

As depicted in Figure 10, the results include ten representative RSI samples from the most confusing categories. The first and third rows display the original images, while the second and fourth rows show the corresponding CAMs. The top five images are from the AID30 dataset, while the bottom five are from the NWPU45 dataset.

In the AID30 samples, the key activation areas, highlighted as brighter regions in the CAMs, are closely associated with ground objects that contribute to the semantic understanding of the scene. For instance, the flat-and-round structures represent the center, recreational facilities represent the park, swimming pools represent the resort, and playgrounds represent the school. Similarly, in the NWPU45 samples, the activation areas emphasize important ground objects that align with human cognitive logic. For example, round structures represent churches, blue structures represent industrial areas, bodies of water represent lakes, and white buildings represent railway stations.

These Grad-CAM visualizations confirm that the features our ERKT-Net learned through the knowledge transfer process are both effective and interpretable.

4.4.2. t-SNE Results

The t-SNE results for the AID30 and NWPU45 datasets are presented in Figures 11 and 12, respectively. The t-SNE technique projects various categories onto a two-dimensional map based on the spatial distances between samples. This allows us to evaluate the effectiveness of the model's features through the separation between categories.

As depicted in Figure 11, all categories in the AID30 dataset are well-separated, with the exception of three pairs of categories, which are marked with red rectangles and have overlapping areas. These overlapping pairs are playground and stadium, commercial and school, and park and resort.

In Figure 12, most categories within the NWPU45 dataset are clearly separated, except for four pairs: the church and palace, the dense residential and medium residential, the railway and railway station, and the lake and wetland. When compared with the previously shown confusion matrix results, these overlapping pairs are consistent.

These, these t-SNE results further confirm that the features learned by our ERKT-Net through the knowledge transfer process are effective.

Table 8: Comparison of Parameters and Inference Time for Various Models on AID30

Model	Params(M)	FLOPs(G)	Inferring time(second)
ResNet-18	11.7	1.8	5.49 ± 0.07
ResNet-50	25.6	4.1	16.50 ± 0.05
DenseNet-121	8.0	2.9	18.75 ± 0.07
MobileNetV2	3.5	0.3	6.56 ± 0.06
EfficientNet-B3	12.1	1.8	15.40 ± 0.05
Ensemble Teacher	43.1	6.3	38.06 ± 0.08
ERKT-Net	5.3	0.4	7.94 ± 0.08

Table 9: Comparison of OA (%) for Various Methods Across Different Training Epochs

Training method	NWPU45 TR10%		NWPU45 TR20%	
	600 epochs	3000 epochs	600 epochs	3000 epochs
Base	93.58 ± 0.11	×	95.01 ± 0.07	×
Original DIST	94.54 ± 0.01	94.81 ± 0.04	96.05 ± 0.02	96.18 ± 0.18
Our ERKT-Net	**94.90 ± 0.05**	×	**96.36± 0.05**	×

4.5. Computational Efficiency Analysis

Apart from model size, inference speed is another crucial metric for performance evaluation during deployment. The total number of convolution operations significantly influences the inference latency of CNNs, particularly when running on a GPU [68]. Consequently, we compared the prediction speed of ERKT-Net with that of the teacher ensemble and other classical CNNs using 6,300 samples from NWPU45.

As shown in Table 8, our ERKT-Net exhibited an inference speed comparable to ResNet-18 and MobileNetV2, while its size is only 45% of that of ResNet-18. Given that ResNet-18 and MobileNetV2 have much lower accuracy on ImageNet-1K than EfficientNet-B0, these results suggest that ERKT-Net offers a superior balance between model size, inference latency, and accuracy.

Compared to the teacher ensemble, our ERKT-Net reduces inference time and model size by 80% and 88%, respectively. This result underscores the efficiency of ERKT-Net.

4.6. Ablation Experiments

In this section, we conducted a series of ablation experiments on NWPU45 to validate the effectiveness of our KD methods. The results are presented in Table 9.

Initially, we established the effectiveness of transfer learning as a baseline by training our student model without the knowledge transfer process. Specifically, we reverted the functions involving VSM to their default state with the following settings: CutMix probability set to 0.1, random horizontal and vertical flip probabilities set to 0.5, and both random erasing and random resized crop deactivated. Using these usual data augmentation strategies presented in the literature [4, 10], we then evaluated the efficiency of the original DIST loss for distilling knowledge over an extended training period, up to 3000 epochs.

As demonstrated in Table 9, the experimental results reveal that the original DIST, even with a much longer training scheme of up to 3000 epochs, cannot match the efficiency of our ERKT-Net. These ablation experiments confirm that ERKT-Net is a more robust and efficient KD approach for achieving lightweight RSI classifiers. Notably, our method, leveraging the inherent nature of RSIs, reduces the time expenditure in the distillation phase by at least 80% compared to the original DIST developed on ImageNet-1K.

5. Conclusions

In this paper, we introduce a novel approach for generating lightweight RSI classifiers using KD techniques. This method presents innovative yet straightforward concepts to better accommodate the inherent nature of RSIs, significantly enhancing the efficiency and robustness of traditional KD techniques developed on ImageNet-1K.

The advantages of our method primarily stem from two aspects. Firstly, we propose a straightforward algorithm for generating a robust three-CNN ensemble as the teacher model. Secondly, we propose a novel VS strategy to address the large variances in data distribution, which are characteristic of RSIs caused by noisy backgrounds and significant similarities across categories.

We evaluated our student model on three benchmark RSI datasets. The results revealed that our ERKT-Net demonstrated superior accuracy and a very compact model size compared to 40 other SOTA methods published between 2020 and 2023. In the challenging NWPU45 dataset, ERKT-Net surpassed other KD-based methods with a maximum OA improvement of 22.4%. Under the same criterion, ERKT-Net also surpassed the top-ranked multi-model method with a minimum OA improvement of 0.6%, using only up to 4.7% of the parameters. Additionally, ablation experiments indicated that our VS strategy, tailored to the inherent nature of RSIs, significantly improved the efficiency and robustness of classic DA techniques for knowledge transfer. Notably, it reduced the time expenditure in the distillation phase by

at least 80% with only a slight accuracy sacrifice.

However, our work is still preliminary and limited, with many aspects requiring improvement in the future. Firstly, we have not conducted a grid search for hyperparameters during the KD process, which may significantly improve the student model's performance with an optimized strategy. Secondly, we have not extensively utilized the spatial and location characteristics of RSIs, such as spatial distance or long-range dependence between ground features, to design more tailored and efficient distillation techniques. We aim to address these concerns in our future work.

Acknowledgements

This work was funded by Hunan University of Arts and Science (grant number: Geography Subject [2022] 351).

References

[1] Xu C, Du X, Fan X, Giuliani G, Hu Z, Wang W, et al. Cloud-based storage and computing for remote sensing big data: a technical review. International Journal of Digital Earth 2022;15:1417–45. https://doi.org/10.1080/17538947.2022.2115567.

[2] Mountrakis G, Heydari SS. Harvesting the Landsat archive for land cover land use classification using deep neural networks: Comparison with traditional classifiers and multi-sensor benefits. ISPRS Journal of Photogrammetry and Remote Sensing 2023;200:106–19. https://doi.org/10.1016/j.isprsjprs.2023.05.005.

[3] Dimitrovski I, Kitanovski I, Kocev D, Simidjievski N. Current trends in deep learning for Earth Observation: An open-source benchmark arena for image classification. ISPRS Journal of Photogrammetry and Remote Sensing 2023;197:18–35. https://doi.org/10.1016/j.isprsjprs.2023.01.014.

[4] Song H, Zhou Y. Simple is best: A single-CNN method for classifying remote sensing images. NHM 2023;18:1600–29. https://doi.org/10.3934/nhm.2023070.

[5] Song H. MBC-Net: long-range enhanced feature fusion for classifying remote sensing images. IJICC 2024;17:181–209. https://doi.org/10.1108/IJICC-07-2023-0198.

[6] Jamali A, Mahdianpari M, Mohammadimanesh F, Homayouni S. A deep learning framework based on generative adversarial networks and vision transformer for complex wetland classification using limited training samples. International Journal of Applied Earth Observation and Geoinformation 2022;115:103095. https://doi.org/10.1016/j.jag.2022.103095.

[7] Song H, Yuan Y, Ouyang Z, Yang Y, Xiang H. Quantitative regularization in robust vision transformer for remote sensing image classification. The Photogrammetric Record. 2024: Online First. https://doi.org/10.1111/phor.12489.

[8] Yue J, Fang L, Ghamisi P, Xie W, Li J, Chanussot J, et al. Optical Remote Sensing Image Understanding With Weak Supervision: Concepts, methods, and perspectives. IEEE Geosci Remote Sens Mag 2022;10:250–69. https://doi.org/10.1109/MGRS.2022.3161377.

[9] Thoreau R, Achard V, Risser L, Berthelot B, Briottet X. Active Learning for Hyperspectral Image Classification: A comparative review. IEEE Geosci Remote Sens Mag 2022;10:256–78. https://doi.org/10.1109/MGRS.2022.3169947.

[10] Song H. A Leading but Simple Classification Method for Remote Sensing Images. AETiC 2023;7:1–20. https://doi.org/10.33166/AETiC.2023.03.001.

[11] Chen J, Di X, Xu R, Luo H, Qi H, Zhan P, et al. An efficient scheme for in-orbit remote sensing image data retrieval. Future Generation Computer Systems 2024;150:103–14. https://doi.org/10.1016/j.future.2023.08.017.

[12] Wang Y, Zhao C, Dong D, Wang K. Real-time monitoring of insects based on laser remote sensing. Ecological Indicators 2023;151:110302. https://doi.org/10.1016/j.ecolind.2023.110302.

[13] Zhang Z, Liu Q, Liu X, Zhang Y, Du Z, Cao X. PMNet: a multi-branch and multi-scale semantic segmentation approach to water extraction from high-resolution remote sensing images with edge-cloud computing. J Cloud Comp 2024;13:76. https://doi.org/10.1186/s13677-024-00637-5.

[14] 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. https://doi.org/10.3390/s20071999.

[15] Chen Z, Yang J, Feng Z, Chen L. RSCNet: An Efficient Remote Sensing Scene Classification Model Based on Lightweight Convolution Neural Networks. Electronics 2022;11:3727. https://doi.org/10.3390/electronics11223727.

[16] Liang L, Wang G. Efficient recurrent attention network for remote sensing scene classification. IET Image Processing 2021;15:1712–21. https://doi.org/10.1049/ipr2.12139.

[17] Zheng F, Lin S, Zhou W, Huang H. A Lightweight Dual-Branch Swin Transformer for Remote Sensing Scene Classification. Remote Sensing 2023;15:2865. https://doi.org/10.3390/rs15112865.

[18] Alhichri H, Alswayed AS, Bazi Y, Ammour N, Alajlan NA. Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention. IEEE Access 2021;9:14078–94. https://doi.org/10.1109/ACCESS.2021.3051085.

[19] Chen S-B, Wei Q-S, Wang W-Z, Tang J, Luo B, Wang Z-Y. Remote Sensing Scene Classification via Multi-Branch Local Attention Network. IEEE Trans on Image Process 2022;31:99–109. https://doi.org/10.1109/TIP.2021.3127851.

[20] Zhao Z, Li J, Luo Z, Li J, Chen C. Remote Sensing Image Scene Classification Based on an Enhanced Attention Module. IEEE Geosci Remote Sensing Lett 2021;18:1926–30. https://doi.org/10.1109/LGRS.2020.3011405.

[21] Wan H, Chen J, Huang Z, Feng Y, Zhou Z, Liu X, et al. Lightweight Channel Attention and Multiscale Feature Fusion Discrimination for Remote Sensing Scene Classification. IEEE Access 2021;9:94586–600. https://doi.org/10.1109/ACCESS.2021.3093308.

[22] Huang X, Liu F, Cui Y, Chen P, Li L, Li P. Faster and Better: A Lightweight Transformer Network for Remote Sensing Scene Classification. Remote Sensing 2023;15:3645. https://doi.org/10.3390/rs15143645.

[23] Xu C, Zhu G, Shu J. A Lightweight and Robust Lie Group-Convolutional Neural Networks Joint Representation for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–15. https://doi.org/10.1109/TGRS.2020.3048024.

[24] Wang X, Xu H, Yuan L, Wen X. A lightweight and stochastic depth residual attention network for remote

sensing scene classification. IET Image Processing 2023;17:3106–26. https://doi.org/10.1049/ipr2.12836.

[25] Shi C, Zhang X, Sun J, Wang L. Remote Sensing Scene Image Classification Based on Self-Compensating Convolution Neural Network. Remote Sensing 2022;14:545. https://doi.org/10.3390/rs14030545.

[26] Bai L, Liu Q, Li C, Ye Z, Hui M, Jia X. Remote Sensing Image Scene Classification Using Multiscale Feature Fusion Covariance Network With Octave Convolution. IEEE Trans Geosci Remote Sensing 2022;60:1–14. https://doi.org/10.1109/TGRS.2022.3160492.

[27] Zhang W, Jiao L, Liu F, Liu J, Cui Z. LHNet: Laplacian Convolutional Block for Remote Sensing Image Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–13. https://doi.org/10.1109/TGRS.2022.3192321.

[28] Bi Q, Zhou B, Qin K, Ye Q, Xia G-S. All Grains, One Scheme (AGOS): Learning Multigrain Instance Representation for Aerial Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–17. https://doi.org/10.1109/TGRS.2022.3201755.

[29] Guo W, Li S, Yang J, Zhou Z, Liu Y, Lu J, et al. Remote Sensing Image Scene Classification by Multiple Granularity Semantic Learning. IEEE J Sel Top Appl Earth Observations Remote Sensing 2022;15:2546–62. https://doi.org/10.1109/JSTARS.2022.3158703.

[30] Shi A, Li Z, Wang X. A lightweight skip-connected expansion inception network for remote sensing scene classification. Remote Sensing Letters 2023;14:1098–108. https://doi.org/10.1080/2150704X.2023.2266118.

[31] Ao L, Feng K, Sheng K, Zhao H, He X, Chen Z. TPENAS: A Two-Phase Evolutionary Neural Architecture Search for Remote Sensing Image Classification. Remote Sensing 2023;15:2212. https://doi.org/10.3390/rs15082212.

[32] Broni-Bediako C, Murata Y, Mormille LHB, Atsumi M. Searching for CNN Architectures for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–13. https://doi.org/10.1109/TGRS.2021.3097938.

[33] Shen J, Cao B, Zhang C, Wang R, Wang Q. Remote Sensing Scene Classification Based on Attention-Enabled Progressively Searching. IEEE Trans Geosci Remote Sensing 2022;60:1–13. https://doi.org/10.1109/TGRS.2022.3186588.

[34] Cristian Buciluǎ, Rich Caruana, Alexandru Niculescu-Mizil. Model Compression, Philadelphia, Pennsylvania, USA: Association for Computing Machinery; 2006, p. Pages 535-541. https://doi.org/10.1145/1150402.1150464.

[35] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network. arXiv, 2015. Available at: https://doi.org/10.48550/arXiv.1503.02531. Accessed on: May 01, 2024.

[36] Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y. FitNets: Hints for Thin Deep Nets, arXiv,2015. Available at: https://doi.org/10.48550/arXiv.1412.6550. Accessed on: May 01, 2024.

[37] Park W, Kim D, Lu Y, Cho M. Relational Knowledge Distillation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA: IEEE; 2019, p. 3962–71. https://doi.org/10.1109/CVPR.2019.00409.

[38] Zhao B, Cui Q, Song R, Qiu Y, Liang J. Decoupled Knowledge Distillation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA: IEEE; 2022, p. 11943–52. https://doi.org/10.1109/CVPR52688.2022.01165.

[39] Huang T, You S, Wang F, Qian C, Xu C. Knowledge Distillation from A Stronger Teacher. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc.; 2022, p. 33716–27. Available at: https://proceedings.neurips.cc/paper_files/paper/2022/file/da669dfd3c36c93905a17ddba01eef06-Paper-Conference.pdf. Accessed on: May 01, 2024.

[40] Yim J, Joo D, Bae J, Kim J. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI: IEEE; 2017, p. 7130–8. https://doi.org/10.1109/CVPR.2017.754.

[41] Stanton S, Izmailov P, Kirichenko P, Alemi AA, Wilson AG. Does Knowledge Distillation Really Work? In: Ranzato M, Beygelzimer A, Dauphin Y, Liang PS, Vaughan JW, editors. Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc.; 2021, p. 6906–19. Available at: https://proceedings.neurips.cc/paper_files/paper/2021/file/376c6b9ff3bedbbea56751a84fffc10c-Paper.pdf. Accessed on: May 01, 2024.

[42] Beyer L, Zhai X, Royer A, Markeeva L, Anil R, Kolesnikov A. Knowledge distillation: A good teacher is patient and consistent. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA: IEEE; 2022, p. 10915–24. https://doi.org/10.1109/CVPR52688.2022.01065.

[43] Chen G, Zhang X, Tan X, Cheng Y, Dai F, Zhu K, et al. Training Small Networks for Scene Classification of Remote Sensing Images via Knowledge Distillation. Remote Sensing 2018;10:719. https://doi.org/10.3390/rs10050719.

[44] Xu K, Deng P, Huang H. Vision Transformer: An Excellent Teacher for Guiding Small Networks in Remote Sensing Image Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–15. https://doi.org/10.1109/TGRS.2022.3152566.

[45] Wang X, Zhu J, Yan Z, Zhang Z, Zhang Y, Chen Y, et al. LaST: Label-Free Self-Distillation Contrastive Learning With Transformer Architecture for Remote Sensing Image Scene Classification. IEEE Geosci Remote Sensing Lett 2022;19:1–5. https://doi.org/10.1109/LGRS.2022.3185088.

[46] Li D, Nan Y, Liu Y. Remote Sensing Image Scene Classification Model Based on Dual Knowledge Distillation. IEEE Geosci Remote Sensing Lett 2022;19:1–5. https://doi.org/10.1109/LGRS.2022.3208904.

[47] Hu Y, Huang X, Luo X, Han J, Cao X, Zhang J. Variational Self-Distillation for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–13. https://doi.org/10.1109/TGRS.2022.3194549.

[48] Xing S, Xing J, Ju J, Hou Q, Ding X. Collaborative Consistent Knowledge Distillation Framework for Remote Sensing Image Scene Classification Network. Remote Sensing 2022;14:5186. https://doi.org/10.3390/rs14205186.

[49] Zhao Q, Ma Y, Lyu S, Chen L. Embedded Self-Distillation in Compact Multibranch Ensemble Network for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–15. https://doi.org/10.1109/TGRS.2021.3126770.

[50] Song H. A Consistent Mistake in Remote Sensing Images' Classification Literature. Intelligent Automation & Soft Computing 2023;37:1381–98. https://doi.org/10.32604/iasc.2023.039315.
[51] Zhang J, Zhao H, Li J. TRS: Transformers for Remote Sensing Scene Classification. Remote Sensing 2021;13:4143. https://doi.org/10.3390/rs13204143.
[52] Lv P, Wu W, Zhong Y, Du F, Zhang L. SCViT: A Spatial-Channel Feature Preserving Vision Transformer for Remote Sensing Image Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–12. https://doi.org/10.1109/TGRS.2022.3157671.
[53] Wang D, Zhang J, Du B, Xia G-S, Tao D. An Empirical Study of Remote Sensing Pretraining. IEEE Trans Geosci Remote Sensing 2023;61:1–20. https://doi.org/10.1109/TGRS.2022.3176603.
[54] Li B, Guo Y, Yang J, Wang L, Wang Y, An W. Gated Recurrent Multiattention Network for VHR Remote Sensing Image Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–13. https://doi.org/10.1109/TGRS.2021.3093914.
[55] Shen J, Yu T, Yang H, Wang R, Wang Q. An Attention Cascade Global–Local Network for Remote Sensing Scene Classification. Remote Sensing 2022;14:2042. https://doi.org/10.3390/rs14092042.
[56] Tang X, Ma Q, Zhang X, Liu F, Ma J, Jiao L. Attention Consistent Network for Remote Sensing Scene Classification. IEEE J Sel Top Appl Earth Observations Remote Sensing 2021;14:2030–45. https://doi.org/10.1109/JSTARS.2021.3051569.
[57] Wang W, Chen Y, Ghamisi P. Transferring CNN With Adaptive Learning for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2022;60:1–18. https://doi.org/10.1109/TGRS.2022.3190934.
[58] Xu K, Huang H, Deng P. Remote Sensing Image Scene Classification Based on Global–Local Dual-Branch Structure Model. IEEE Geosci Remote Sensing Lett 2022;19:1–5. https://doi.org/10.1109/LGRS.2021.3075712.
[59] Deng P, Xu K, Huang H. When CNNs Meet Vision Transformer: A Joint Framework for Remote Sensing Scene Classification. IEEE Geosci Remote Sensing Lett 2022;19:1–5. https://doi.org/10.1109/LGRS.2021.3109061.
[60] Zhao M, Meng Q, Zhang L, Hu X, Bruzzone L. Local and Long-Range Collaborative Learning for Remote Sensing Scene Classification. IEEE Trans Geosci Remote Sensing 2023;61:1–15. https://doi.org/10.1109/TGRS.2023.3265346.
[61] Ma J, Li M, Tang X, Zhang X, Liu F, Jiao L. Homo–Heterogenous Transformer Learning Framework for RS Scene Classification. IEEE J Sel Top Appl Earth Observations Remote Sensing 2022;15:2223–39. https://doi.org/10.1109/JSTARS.2022.3155665.
[62] Wang G, Chen H, Chen L, Zhuang Y, Zhang S, Zhang T, et al. P2FEViT: Plug-and-Play CNN Feature Embedded Hybrid Vision Transformer for Remote Sensing Image Classification. Remote Sensing 2023;15:1773. https://doi.org/10.3390/rs15071773.
[63] Cheng X, Lei H. Remote Sensing Scene Image Classification Based on mmsCNN–HMM with Stacking Ensemble Model. Remote Sensing 2022;14:4423. https://doi.org/10.3390/rs14174423.
[64] Sesmero MP, Ledezma AI, Sanchis A. Generating ensembles of heterogeneous classifiers using Stacked Generalization. WIREs Data Min & Knowl 2015;5:21–34. https://doi.org/10.1002/widm.1143.
[65] Yun S, Han D, Chun S, Oh SJ, Yoo Y, Choe J. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, p. 6022–31. https://doi.org/10.1109/ICCV.2019.00612.
[66] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Int J Comput Vis 2020;128:336–59. https://doi.org/10.1007/s11263-019-01228-7.
[67] Maaten L van der, Hinton G. Visualizing Data using t-SNE. Journal of Machine Learning Research 2008;9:2579–605. Available at: http://jmlr.org/papers/v9/vandermaaten08a.html. Accessed on: May 01, 2024.
[68] Radosavovic I, Kosaraju RP, Girshick R, He K, Dollar P. Designing Network Design Spaces. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE; 2020, p. 10425–33. https://doi.org/10.1109/CVPR42600.2020.01044.