Class-Aware Self-Distillation for Remote Sensing Image Scene Classification

Bin Wu, Siyuan Hao, *Member, IEEE*, and Wei Wang

***Abstract*—Currently, convolutional neural networks (CNNs) and vision transformers (ViTs) are widely adopted as the predominant neural network architectures for remote sensing image scene classification. Although CNNs have lower computational complexity, ViTs have a higher performance ceiling, making both suitable as backbone networks for remote sensing scene classification tasks. However, remote sensing imagery has high intraclass variation and interclass similarity, which poses a challenge for existing methods. To address this issue, we propose the class-aware self-distillation (CASD) framework. This framework uses an end-to-end distillation mechanism to mine class-aware knowledge, effectively reducing the impact of significant intraclass variation and interclass similarity in remote sensing imagery. Specifically, our approach involves constructing pairs of images: similar pairs consisting of images belonging to the same class, and dissimilar pairs consisting of images from different classes. We then apply a distillation loss that we designed, which distills the corresponding probability distributions to ensure that the distributions of similar pairs become more consistent, and those of dissimilar pairs become more distinct. In addition, the enforced learnable α added to the distillation loss further amplifies the network's ability to comprehend class-aware knowledge. The experiment section demonstrates that our method CASD outperforms other methods on four publicly available datasets. And the ablation experiments demonstrate the effectiveness of the method.**

***Index Terms*—Deep learning, knowledge distillation (KD), remote sensing image, scene classification, vision transformer (ViT).**

I. Introduction

IN THE modern technological era, high spatial resolution (HSR) remote sensing images are increasingly utilized in various sectors. These applications include disaster detection [1], [2], geographic object detection [3], [4], and land-use classification [5], [6], [7], [8]. An abundance of research focuses on understanding HSR remote sensing images. Remote sensing image scene classification refers to categorizing images into predefined land-cover/land-use classes. It is a key topic in aerial and satellite image analysis and essential for understanding HSR images. Yet, classifying these images is challenging due to the complex relationships among ground objects.

In recent scientific advancements, the development of scene classification methods has garnered widespread attention in the field of computer vision. These methods can be broadly divided into two main categories: hand-crafted feature-based methods [9], [10], [11], [12], [13], [14] and deep-learning-based methods. The hand-crafted feature-based method is less commonly used nowadays due to its requirement for specific expert knowledge and its inability to achieve efficient end-to-end classification. In contrast, deep-learning-based methods offer an alternative approach by automatically learning to extract relevant features through the neural network architecture, resulting in improved performance and reduced workload.

Deep learning-based methods include deep belief nets, stacked autoencoders, convolutional neural networks (CNNs), and vision transformers (ViTs). Among these, the CNN is the most widely used network structure for remote sensing image scene classification tasks. Recently, the ViT has emerged as a popular and innovative approach in the field of computer vision.

There are two key differences between ViTs and CNNs for the classification task:

1) Unlike the local convolutional strategy of CNNs, ViTs divide images into a sequence of patches and model the global relationships between patches using the self-attention mechanism. This allows ViTs to capture long-range dependencies and abstract higher order semantic labels for remote sensing images with complex spatial layouts. However, modeling global relationships also increases the computational complexity of ViTs, reducing the computational efficiency.
2) CNNs rely on their own inductive bias (e.g., locality and spatial invariance) and perform well on limited data. However, this inductive bias may limit the information learned when there is ample data available [15]. On the other hand, ViTs rely on a more flexible self-attention mechanism, making it easier to achieve a higher performance ceiling compared to CNNs.

In summary, CNNs are computationally efficient and do not heavily depend on pretrained data, while ViTs are computationally intensive but can attain higher performance with sufficient pretrained data.

Manuscript received 9 August 2023; revised 30 October 2023 and 20 November 2023; accepted 12 December 2023. Date of publication 15 December 2023; date of current version 3 January 2024. This work was supported in part by the National Natural Science Foundation of China, specifically under Grant K23I00040, Grant 62171247, and Grant 41921781. *(Corresponding author: Siyuan Hao.)*

Bin Wu is with the College of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China (e-mail: wubin970623@gmail.com).

Siyuan Hao is with the School of Software Engineering, Beijing Jiaotong University, Beijing 100091, China (e-mail: lemonbananan@163.com).

Wei Wang is with the Institute of Information Science, Beijing Jiaotong University, Beijing 100091, China (e-mail: wei.wang@bjtu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3343521

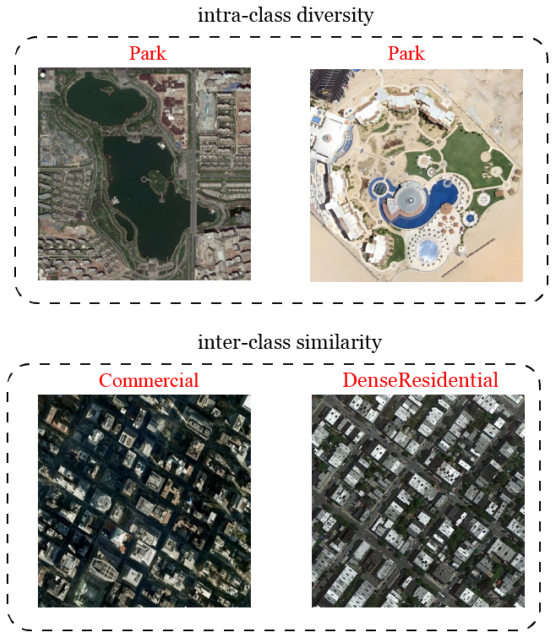


Fig. 1. Examples illustrating intraclass diversity and interclass similarity of remote sensing images.

Remote sensing images present unique challenges due to their complex scene layouts and varied ground object information. The complexity results in significant intraclass diversity, meaning images of the same scene can vary considerably. This is largely due to differences in physical conditions, seasonal changes, and the imaging sensor used. Further complicating matters is interclass similarity resulting from semantic overlap between different scenes. For instance, an urban park and a rural forest might appear similar due to shared features, causing potential confusion in classification. These factors make the classification task in remote sensing image analysis particularly challenging. Fig. 1 shows that the "Park" images have high intraclass diversity. For instance, there are significant differences in the scene layouts, colors, and textures between the two "Park" images. This makes it hard to be classified correctly as the training and testing images might be visually quite different. In contrast, the "Commercial" and "DenseResidential" images have interclass similarity due to their similar building shapes and scene layouts. This makes them hard to be distinguished from each other. As a result, how to classify these images that have high intraclass diversity and interclass similarity is a big challenge.

To address this challenge, for remotely sensing images, it is expected that pairs of the same class will be closer, while pairs of different classes are separated in the feature space. This results in a decrease in intraclass distance and an increase in interclass distance, effectively improving the classification performance. Therefore, the model must be trained to acquire additional knowledge to achieve this objective. Knowledge distillation (KD) is a training method that enables student models to grow by learning the output distribution of a teacher network. While some self-distillation methods can achieve the transfer of class-aware knowledge, they often only focus on intraclass knowledge or interclass knowledge, and cannot effectively combine the two. Previously, Zhang et al. [16] proposed a self-distillation framework to learn class information by making the output distribution of the same class images more consistent. However, this only reduced the intraclass distance and did not address the interclass distance. To tackle both distances, we proposed the class-aware self-distillation (CASD) framework.

The CASD framework is based on the principle of KD, where the objective is to transfer knowledge from a teacher network to a student network. However, in the case of CASD, both the teachers and the student are the same network. There are two teachers in CASD, with one imparting the similarity of samples within the same class and the other imparting the difference between samples from different classes to the student. Through this process, the student network gains class-aware knowledge to improve the performance of the classification model. Furthermore, we have constructed a unique distillation loss for the transfer of knowledge, which includes a learnable interval α to amplify the class-aware knowledge.

Our experiments on four benchmark datasets using ResNet and ViT architectures show that the proposed CASD method has good generalization and improved classification performance.

In summary, we make the following principle contributions in this article.

1) A CASD framework is proposed for the remote sensing image scene classification. The CASD framework cleverly utilizes self-distillation to simultaneously alleviate the issue of intraclass diversity and interclass similarity in remote sensing images by extracting class-aware information.
2) Our constructed class distillation loss enables the transfer of class-aware knowledge both intraclass and interclass. More importantly, through a learnable interval, it ensures that the network can adaptively adjust interclass distances, significantly enhancing the model's performance.
3) Experimental results show that our method has good generalizability. Both the ResNet and ViT architectures achieved significant performance enhancement on four benchmark datasets of remote sensing images.

II. Related Work

A. CNN-Based Methods

Some CNN-based methods will be simply enumerated. Inspired by visual attention mechanisms, Wang et al. [17] first introduced the attention mechanism to remote sensing image classification. In 2019, Zhang et al. [18] proposed CNN-Capsnet, which combines the CNN with a capsule network (CapsNet). Subsequently, Wang et al. [19] present a global-local two-stream architecture to address the large-scale variation of features and objects in remote sensing images, achieving state-of-the-art results on four public datasets. Xu et al. [20] introduced a graph-convolutional-network-based model that effectively captures context relationships and refines features for high spatial resolution scene classification, outperforming several state-of-the-art methods. Wan et al. [21] proposes an efficient multi-objective evolutionary framework that balances interpretation

accuracy and parameter quantity for remote sensing image scene classification, demonstrating the effectiveness of the approach in comparison to human-designed networks and other search methods. Another interesting work is by Xu et al. [22], who effectively combined lie group machine learning with the CNN to enhance the expressive power of the CNN. Some methods are dedicated to analyzing and reducing the dimensionality of remote sensing data to enhance the model performance. Makantasis et al. [23], [24] mainly focus on how to capture the multidimensional structure of data through the tensor method, and build a more effective and accurate data analysis model.

B. ViT-Based Methods

As an attention-based structure model, the transformer [25], [26], [27] demonstrated tremendous force in sequence modeling and machine translation with great success. Researchers have tried to transplant transformers to the field of computer vision inspired by the successful application of transformers in natural language processing. In 2020, Dosovitskiy et al. [28] proposed the ViT by using image patches as input for image classification, which boosted the classification performance of SOTA. Since a ViT lacks some inductive biases inherent to CNNs (such as translation invariance and localization), it cannot generalize well when the trained data are insufficient. However, when pretrained at sufficient scale, the ViT can achieve excellent results on downstream tasks with a less amount of data.

The ViT already has many applications in scene classification with its powerful image recognition capability. In 2021, Bazi et al. [29] introduced a pruned ViT to remote sensing image scene classification and explored the effect of different data augmentation strategies on the ViT. Kaselimi et al. [30] proposed a multilabel ViT called ForestViT to solve the problem of satellite image classification regarding deforestation monitoring. The interaction of the CNN and ViT has also been studied recently. CTNet was proposed by Deng et al. [31], it uses both CNN and ViT streams concurrently, complementing semantic information with local structural information, thereby extracting more distinctive features. Subsequently, TRSNet [32] was proposed, which enhanced the capacity of the ViT by integrating it with the CNN in a serial structure. While the combination of the CNN and ViT can certainly improve the classification performance, how effective are pure ViT structure models in the task of remote sensing image scene classification? The SCViT proposed by Lv et al. [33] fully exploits the spatial and spectral information and demonstrates that a pure ViT may also achieve better classification performance. In addition, Hao et al. [34] proposed TSTNet, which improves the performance of the Swin transformer in the field of scene classification using edge information as a priori.

C. Knowledge Distillation (KD)

KD, a concept introduced by Hinton et al. [35], is a model compression method that facilitates the transfer of knowledge from a larger, well-trained model, often referred to as the teacher, to a smaller one, known as the student. This process enhances the learning capabilities of the student model, enabling it to mimic the performance of its larger counterpart, although with a significantly reduced computational footprint. A variant of this method, self-distillation, innovatively eliminates the need for a separate teacher model. Instead, it employs the same network for both student and teacher roles, essentially using the model's own predictions as supervisory signals. Zhang et al. [36], [37] proposed two self-distillation methods that transfer knowledge from deeper to shallower sections of the network, reinforcing the idea of self-distillation as an effective technique to enhance the model performance. In the realm of remote sensing image processing, the application of self-distillation has proven to be highly beneficial. It not only improves model robustness but also enhances feature learning ability, aiding in more accurate and efficient interpretation of complex remote sensing data. In remote sensing image processing, self-distillation is used to improve model robustness and feature learning ability. Wang et al. [38] used self-distillation to transfer knowledge from ensemble branches to the main branch, reducing the network complexity. Duan et al. [39] designed a self-context distillation module, and Hu et al. [40] proposed a variational self-distillation network with a variational knowledge transfer module.

In contrast to previous works, our objective is not to learn knowledge in a general sense, but rather class-aware information. To achieve this, we have designed a single network that takes three inputs: an anchor, a positive sample (a sample from the same class as the anchor), and a negative sample (a sample from a different class than the anchor). After the network has learned the features of the three inputs in feature space, the loss function we have designed ensures that the distribution of the anchors and positive samples are similar, while the distribution of the anchors and negative samples are dissimilar. This approach effectively mitigates the issue of high intraclass diversity and interclass similarity in remotely sensing images.

III. Proposed Method

Fig. 2 shows the general architecture of CASD. This section focuses on the following three parts: 1) backbone networks; 2) three-branch distillation; and 3) class-aware distillation loss. The specific details of CASD are provided in Sections III-A–III-C.

A. Backbone Networks

Both ViT and CNN play significant roles in scene classification. ViT, with its robust global modeling capability, offers superior performance, while CNNs, exemplified by ResNet-50, bring speed advantages. In real-world applications, one should choose flexibly based on the balance between speed and performance. Accordingly, we developed two versions of CASD: CASD-ViT and CASD-ResNet50. Currently, compared to the well-established CNN architectures, there is limited research on the self-distillation of the ViT in terms of categories. Therefore, CASD-ViT is our primary focus and best demonstrates the performance potential of CASD. In Fig. 2, f_θ refers to the backbone network for extracting features, such as ResNet or ViT, and g_θ refers to a two-layer fully connected multilayer perception (MLP) network.

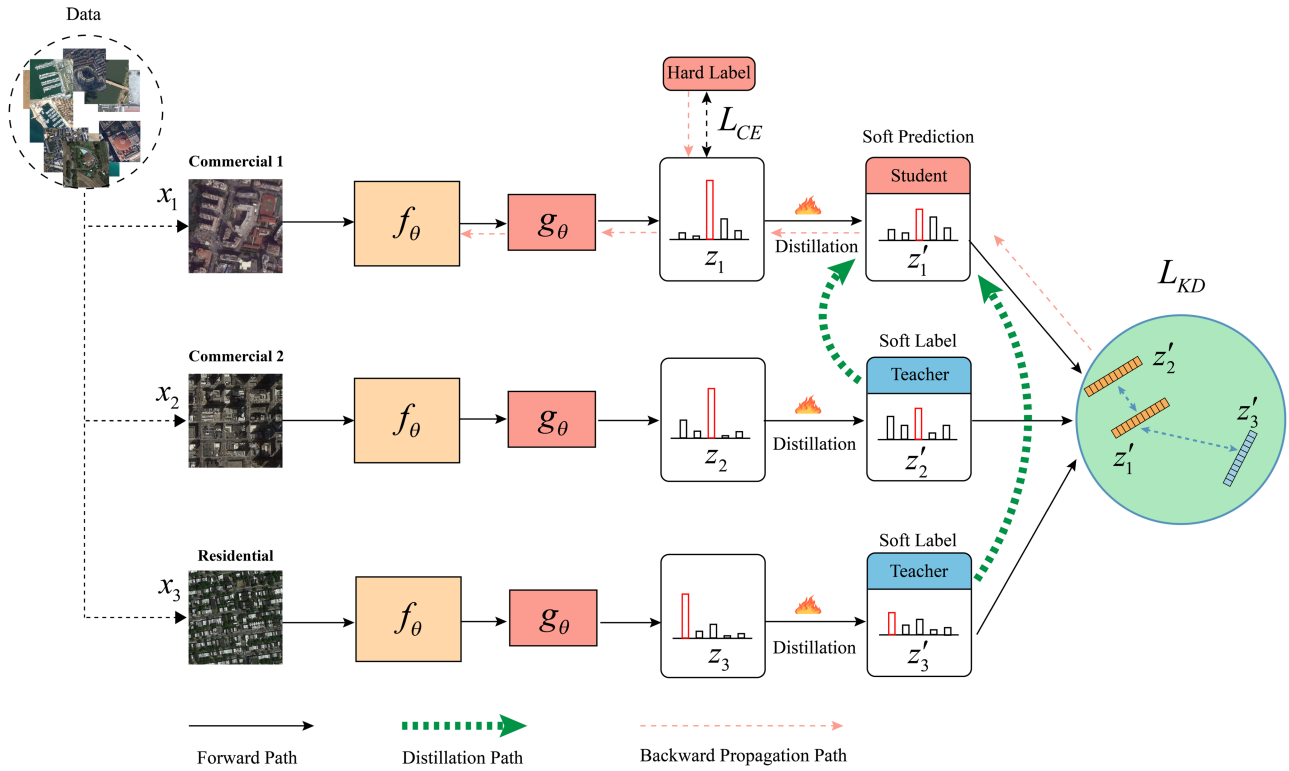


Fig. 2. Framework of the CASD. CASD differs from standard KD in that its three branches networks share weights. During the training process, samples from the same class (e.g., Commercial 1 and Commercial 2) or different classes (e.g., Commercial 1 and Residential) are input into the feature extraction network f_θ. The output logits z are then obtained after being transformed through the linear projection network g_θ, which is a two-layer MLP network. The logits z are distilled into a soft probability distribution z' through a temperature T. The class-aware knowledge is transferred from the two teacher branches to the student branch through the distillation loss L_{KD}. Both L_{KD} and L_{CE} work in tandem to effectively reduce the intraclass distance and increase the interclass distance. It is important to note that CASD is only used during the training stage, and only the backbone network is retained for validation.

First, we introduce the feature extraction process of the ViT model. ViT's input is sequence data. First, the input remote sensing image $I \in \mathbb{R}^{H\times W\times 3}$ is divided into multiple patches (also called tokens), where $H \times W$ denotes the height and width of the three-channel image. The image $I \in \mathbb{R}^{H\times W\times 3}$ is converted into sequence data $T_\partial \in \mathbb{R}^{N\times D}$ after reshape operation and linear projection, where N is the number of patches, and D refers to the mapping of each patch to a D-dimensional space. Since the ViT needs to learn the location information of each token, position embedding $P_{\mathrm{emb}} \in \mathbb{R}^{N\times D}$ is added to the input sequence T_∂. The input of the ViT is formulated as follows:

$$T_0 = T_\partial + P_{\mathrm{emb}}. \tag{1}$$

After the sequence is constructed, it is fed into the ViT block to achieve the information interaction between tokens. Each ViT block consists of a multihead self-attention (MSA) and an MLP

$$\begin{aligned} T'_l &= \mathrm{MSA}(\mathrm{LN}(T_{l-1})) + T_{l-1}, \quad l = 1, \ldots, L \\ T_l &= \mathrm{MLP}(\mathrm{LN}(T'_l)) + T'_l, \qquad l = 1, \ldots, L \end{aligned} \tag{2}$$

where T_{l-1} is the output tokens of block $l-1$, T'_l and T_l are the outputs of MSA and MLP respectively, and LN is the layer normalization. The output $T_L \in \mathbb{R}^{N\times D}$ of the ViT encoder is fed to the global average pooling (GAP) to generate feature vector $\mathrm{h} \in \mathbb{R}^{1\times D}$.

The CASD framework has been applied not only to the ViT but also to the CNN, specifically the ResNet-50 architecture. In this implementation, ResNet-50 is chosen as the other backbone due to its shallow structure and relatively small number of parameters. The structure of ResNet-50 is on the right side of Fig. 3. The remote sensing image $I \in \mathbb{R}^{H\times W\times 3}$, after being fed into continuous residual blocks, yields the feature $F \in \mathbb{R}^{H'\times W'\times C}$. Subsequently, the feature F is dimensionally reduced to $h \in \mathbb{R}^{1\times C}$ by GAP, as described by the following equation:

$$h = \frac{1}{H' \times W'} \sum_{m=1}^{H'} \sum_{n=1}^{W'} F_{m,n,l} \qquad (l = 1, 2, \cdots C) \tag{3}$$

where W' and H' are the feature sizes and C is the number of feature channels.

In addition, we designed g_θ after the feature extraction f_θ. g_θ consists of a two-layer fully connected network and an ReLU activation function as shown in the bottom of Fig. 3. The purpose of g_θ is to map the representation from the feature space to the semantic space. g_θ receives the feature vector h from the feature extraction network f_θ to deliver the higher order semantic information and generate the semantic feature vector $z \in \mathbb{R}^c$, where c is the number of classes. The process is formulated

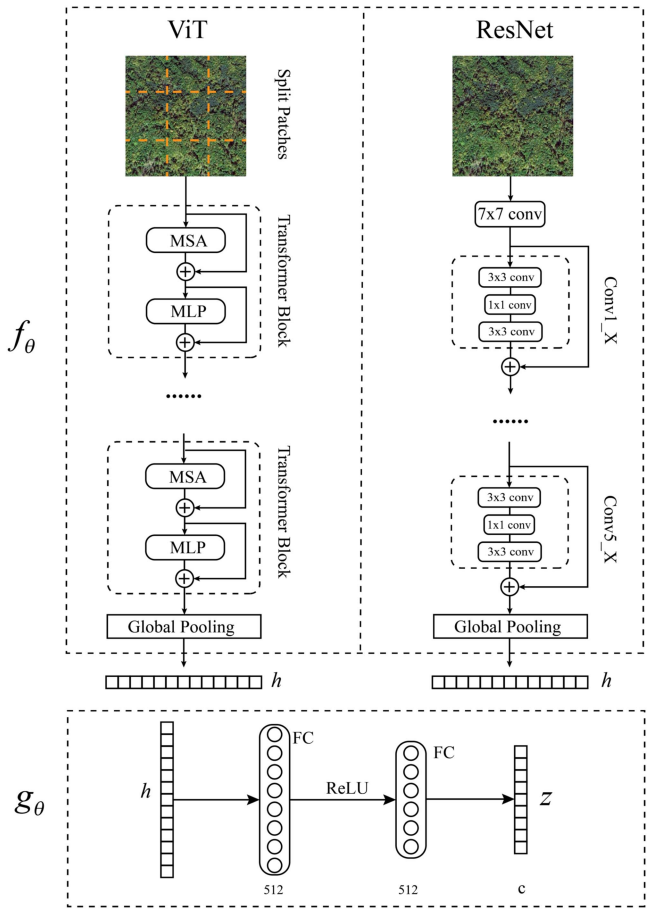


Fig. 3. Structure of the feature extraction network. f_θ refers to ViT or ResNet. g_θ maps the feature embedding h into logits z.

as follows:

$$z = \mathrm{FC}_1(\mathrm{ReLU}(\mathrm{FC}_2(h))). \tag{4}$$

B. Three-Branch Distillation

CASD is an end-to-end self-distillation method aimed at enhancing classification performance by decreasing intraclass distance and increasing interclass distance. The cross-entropy loss is widely employed for multiclass classification tasks and it has the following formula:

$$L_{\mathrm{CE}} = \sum_{c=1}^{M} y_c \log(p_c) \tag{5}$$

where M is the total number of classes and p_c is the confidence of the cth class. y_c is the label code of the cth class, which has the value of 1 or 0. Observing the labels of one-hot form $[y_1, y_2 \cdots y_M]$, we calculate their corresponding values only for the correctly predicted class. For the other incorrectly predicted classes, discarding was performed. Therefore, the cross-entropy loses some of the information entropy. In fact, the labels used for cross-entropy loss are called hard labels. The high-temperature softmax function improves the information entropy of the model output distribution. It not only makes the output logits transformed into a posterior probability distribution but also softens the probability distribution compared to the regular softmax. The z to z' in Fig. 2 shows the softening process of softmax with temperature T. Given an input x and the ground-truth label $y \in Y = 1, \ldots, c$, we can express the predictive distribution Q as

$$Q(y|x;\theta,T) = \frac{\mathrm{e}^{(f_y(x;\theta)/T)}}{Z}. \tag{6}$$

Here, f_y stands for the logit of the model for class y, parameterized by θ, and $T > 0$ denotes the temperature. Z is a normalizing factor defined as

$$Z = \sum_{j=1}^{c} \mathrm{e}^{f_j(x;\theta)/T}. \tag{7}$$

The aforementioned high-temperature softmax function is simplified to $\sigma(\cdot)$

CASD has three branches receiving three consecutive inputs x_1, x_2, and x_3, where x_1 and x_2 have the same label as similar pairs, and x_1 and x_3 have different labels as dissimilar pairs. The three branch networks are the same network and share the same weights, thus being defined as a self-distillation framework. For the ternary input, the feature extraction is denoted as

$$z_i = g_\theta(f_\theta(x_i)) \qquad i = 1, 2, 3. \tag{8}$$

After getting logits $z_1, z_2,$ and z_3, they are fed into the high temperature softmax $\sigma(\cdot)$ to distill more knowledge, the procedure is as follows:

$$z'_i = \sigma(z_i/T) \qquad i = 1, 2, 3 \tag{9}$$

z'_1 is a soft probability prediction, and z'_2 and z'_3 are soft labels containing class-aware information. The first branch is the student network that receives the class-aware knowledge from the teacher network (second and third branches).

C. Class-Aware Distillation Loss

To facilitate the acquisition of class-aware knowledge from z'_2 and z'_3 by z'_1, a suitable metric is necessary to quantify the difference in probability distributions among them. The Kullback–Leibler (KL) divergence is a prevalent metric employed in KD, which we utilize in this article. The KL divergence between probability distributions $p(x)$ and $q(x)$ can be expressed as follows:

$$\mathrm{KL}(p \parallel q) = E_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]. \tag{10}$$

Subsequently, we constructed similar pairs $\langle z'_1, z'_2 \rangle$ and dissimilar pairs $\langle z'_1, z'_3 \rangle$. Reducing the distribution differences of $\langle z'_1, z'_2 \rangle$ enables the model to learn intraclass consistency. Increasing the distribution difference of $\langle z'_1, z'_3 \rangle$ enables the model to learn the diversity of different classes. In addition, we aim to ensure that the difference between $\langle z'_1, z'_3 \rangle$ is significantly greater than that of $\langle z'_1, z'_2 \rangle$, which promotes the extraction of

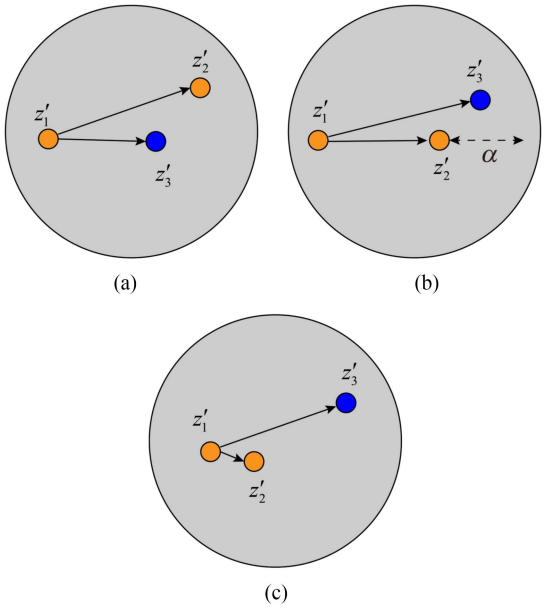


Fig. 4. Three optimization objectives for distillation loss. (a) Intra-class distance is greater than inter-class distance. (b) Intra-class distance is slightly less than inter-class distance. (c) Intra-class distance is much less than inter-class distance.

more discriminative features. Hence, we formulated the proposed mining class distillation loss as follows:

$$L_{\mathrm{KD}} = \frac{1}{N}\sum^{N} \max\{\mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_2) - \mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_3) + \alpha, 0\} \tag{11}$$

where N refers to the number of batches, and $\alpha > 0$ is a learnable interval between similar and dissimilar pairs. Typically, α is set to a specific initial value greater than 0 (e.g., 0.1) and is allowed to update as the network learns, until it reaches an appropriate interval. Such a design usually has two advantages. First, it poses challenges to the network, enhancing the system's robustness. On the other hand, it allows for learning different intervals at different stages of network training until converging to a suitable interval. As illustrated in Fig. 4, the optimization of L_{KD} involves three objectives. The first objective is to

$$\mathrm{KL}(z'_1, z'_2) > \mathrm{KL}(z'_1, z'_3), \qquad L_{\mathrm{KD}} > \alpha. \tag{12}$$

Corresponding to Fig. 4(a), the L_{KD} is larger and needs more optimization.

Second optimization objective corresponds to Fig. 4(b) as follows:

$$\mathrm{KL}(z'_1, z'_2) < \mathrm{KL}(z'_1, z'_3) < \mathrm{KL}(z'_1, z'_2) + \alpha$$
$$0 < L_{\mathrm{KD}} < \alpha. \tag{13}$$

This objective is a simple case where L_{KD} is greater than 0 and less than α, and still needs to be optimized.

We expect $\langle z'_1, z'_3\rangle$ to be much larger than $\langle z'_1, z'_2\rangle$, so we use α as the interval. The third optimization objective corresponds to Fig. 4(c) as follows:

$$\mathrm{KL}(z'_1, z'_2) + \alpha < \mathrm{KL}(z'_1, z'_3), \qquad L_{\mathrm{KD}} = 0 \tag{14}$$

Algorithm 1: CASD Framework.

training stage;
Input: (x_1, x_2, x_3, y); x_1 and x_2 belong to the same class; x_1 and x_3 belong to the different class; y is ground-truth label of x_1;
The parameter θ is initialized ;
while θ *has not converged* **do**
  $z_1 \leftarrow$ model(x_1);
  with torch.no_grad():
    $z_2 \leftarrow$ model(x_2);
    $z_3 \leftarrow$ model(x_3);
  z'_1,z'_2,z'_3 $\leftarrow$ z_1,z_2,z_3; High temperature distilled logits to obtain probability distribution;
  $L_{CE} \leftarrow$ CrossEntropy(z_1, y);
  $L_{KD} \leftarrow \max\{\mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_2) - \mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_3) + \alpha, 0\}$;
  $L \leftarrow \lambda L_{CE} + (1-\lambda)L_{KD}/T^2$;
  Loss L back propagation to update θ;
end

$\langle z'_1, z'_3\rangle$ is much larger than $\langle z'_1, z'_2\rangle$ is an ideal objective, that is, $\mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_2) - \mathrm{KL}(\mathrm{z'}_1, \mathrm{z'}_3) + \alpha < 0$. Subsequently, the loss is 0 by the max function. The third objective does not need to be optimized.

Ultimately, the total loss L is a joint cross-entropy loss L_{CE} and mining class distillation loss L_{KD}:

$$L = \lambda L_{\mathrm{CE}} + (1-\lambda)L_{\mathrm{KD}}/T^2 \tag{15}$$

where λ is the balance coefficient of the two losses and T is the temperature coefficient, which is used to increase the backward gradient of distillation loss. The proposed CASD framework is summarized in Algorithm 1.

IV. Experiments

A. Datasets Description

In the subsequent experiments, we used four publicly available datasets: the NWPU-RESISC45 (NWPU) [41] dataset; Aerial Image dataset (AID) [42]; UC Merced (UCM) [43] dataset; and OPTIMAL-31 [17] datasets; example images of the four datasets are shown in Figs. 5–8. Among them, the NWPU and AID datasets have more scenes, while these scenes have complex spectral and spatial distributions and are the most commonly used benchmark datasets. The UCM and OPTIMAL-31 datasets are relatively small in magnitude and low in a challenge. Among them, the AID, NWPU, and OPTIMAL-31 datasets have high intraclass diversity and interclass similarity, which pose a big challenge for classification methods. Details of the NWPU, AID, UCM, and OPTIMAL-31 datasets are placed in Table I.

B. Experiment Settings

1) Hardware and Software Environment: All experiments were run on a server configured with an Intel(R) Xeon(R) Silver 4214R central processing unit (CPU) at 2.40 GHz and four NVIDIA Geforce RTX 3090 high-speed graphics processing

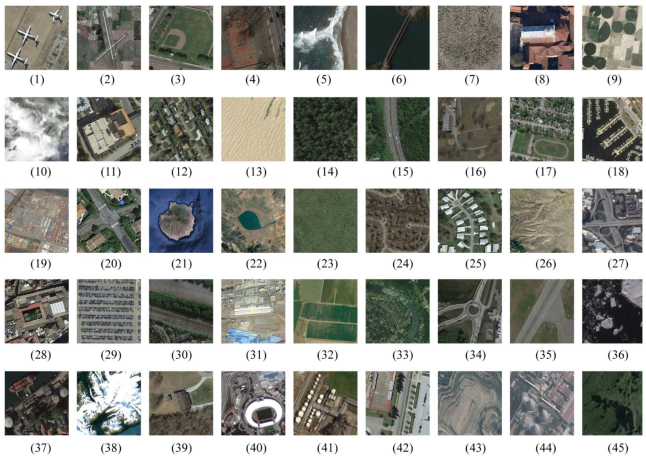


Fig. 5. Example display of 45 scenarios for the NWPU dataset.

Fig. 6. Example display of 30 scenarios for the AID dataset.

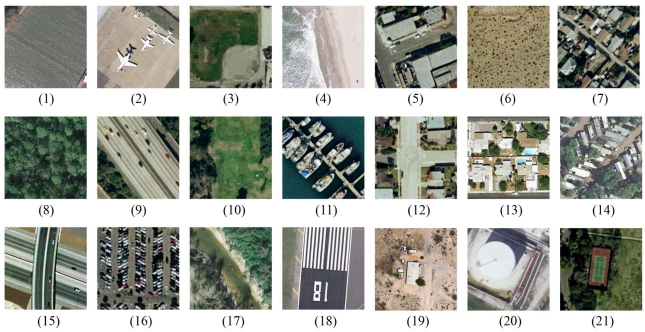


Fig. 7. Example display of 21 scenarios for the UCM dataset.

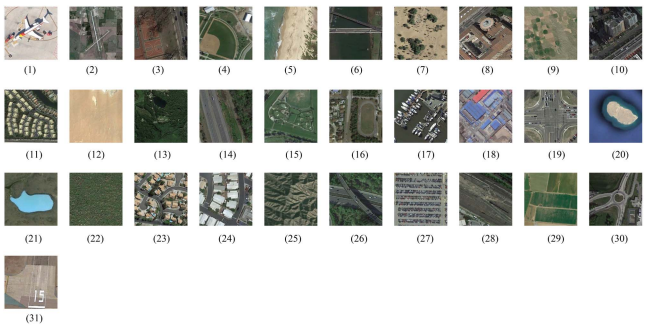


Fig. 8. Example display of 31 scenarios for the OPTIMAL-31 dataset.

TABLE I
DETAILED INFORMATION ON NWPU, AID, UCM, AND PATTERNNET DATASETS

Datasets	NWPU	AID	UCM	OPTIMAL-31
Classes	45	30	21	31
Total images	31500	10000	2100	1860
Images per class	700	220–420	100	60
Image size	256×256	600×600	256×256	256×256
Spatial resolution	0.2-30 m	0.5-8 m	0.3 m	-
Training ratios	10% & 20%	50% & 80%	50% & 80%	80%
Data source	Google Earth	Google Earth	USGS	Google Earth
Release date	2017	2017	2010	2019

units (GPUs) with 24 GB of memory. The software environment for CASD is Ubuntu 18.04.6 LTS with the deep learning framework Pytorch 1.8.0 and Python 3.8.

2) Optimization and Hyperparameters: The backbone of the proposed CASD is loaded with pretrained weights obtained by training on ImageNet-1 K, and then, it is fine tuned on remote sensing images. CASD offers two versions, including the CASD-ViT-B or CASD-ResNet50. The model needs to be trained for 100 epochs, including 15 epochs of warmup strategy. In the training stage, the batch size is set to 64, and we used the Adam optimizer. The learning rate was set to 5e-4. In addition, the cosine learning rate scheduler was used with a minimum learning rate of 1e-6. Finally, we recommend CASD's hyperparameter settings. In the training stage, the distillation temperature coefficient T is set to 5 and learnable interval parameter α is initialized to 0.1 and is constrained to be greater than 0. In the final loss function L, the coefficients λ is set to 0.8.

3) Data Processing and Evaluation Metrics: In subsequent experiments, we will utilize portions of different datasets as training sets. Specifically, for the NWPU dataset, we will employ 10% and 20% of the dataset as the training set. For the AID dataset, we will use 20% and 50% of the dataset for training. With the UCM dataset, we will utilize 50% and 80% of the data as the training set. Finally, for the OPTIMAL-31 dataset, we will use 80% of the data for training purposes. The image size was resized, and the center cropped to 224×224 to be consistent with the previous method on the three datasets. Data augmentation techniques are used in order to improve the generalization of CASD. Data enhancement techniques include horizontal flip, random rotation, automatic contrast, sharpness, etc.

For the task of remote sensing image scene classification, the prevalent validation metrics utilized are overall classification accuracy (OA) and the confusion matrix (CM). In addition, to mitigate the potential influence of dataset partitioning and other random variables, the experiments were conducted five times. Hence, the experimental results are presented as standard deviations to ensure the reliability of the outcomes.

C. Comparison With State-of-The-Arts

We compared our proposed method and the state-of-the-art methods using four widely adopted public benchmark datasets, and employed OA and CM as evaluation metrics. Note that some

TABLE II
ACCURACY (OA ± STD) OF CASD COMPARED WITH OTHER METHODS ON THE NWPU DATASET

Method	Year	10% Train Ratio	20% Train Ratio
CNN-based Methods			
ResNet-50 [44]	2015	90.76±0.11	93.11±0.08
VGG16 [42]	2017	83.59±0.26	87.45±0.18
GoogleNet [42]	2017	81.29±0.28	83.51±0.36
EfficientNet-B0-aux [45]	2019	89.96±0.27	–
EfficientNet-B3-aux [45]	2019	91.08±0.14	–
Inception-v3-CapsNet [18]	2019	89.03±0.21	92.60±0.11
Contourlet CNN [46]	2020	85.93±0.51	89.57±0.45
ResNeXt-101+MTL [47]	2020	91.91±0.18	94.21±0.15
BiMobileNet [48]	2020	92.06±0.14	94.08±0.11
VGG_MS2AP [49]	2021	92.27±0.21	93.91±0.15
CSDS [50]	2021	91.64±0.16	93.59±0.21
GLDBS [51]	2021	92.24±0.21	94.46±0.15
ACNet [52]	2021	91.09±0.13	92.42±0.16
Xu's method [22]	2021	91.91±0.15	94.43±0.16
ViT-based Methods			
V16_21K[224×224] [29]	2021	92.60±0.10	–
ViT [28]	2020	92.15±0.25	94.19±0.10
SCViT [33]	2022	92.72±0.04	94.66±0.10
Distillation-based Methods			
ET-GSNet [53]	2022	92.72±0.28	94.50±0.18
ViT-CL [54]	2022	92.85	94.96
EMSCNet-ResNet50 [55]	2023	92.16±0.07	94.08±0.20
Our Methods			
CASD-ResNet50	2023	92.28±0.23	94.75±0.14
CASD-ViT	2023	**93.12±0.12**	**95.52±0.16**

The bold values represent the highest classification accuracy at the corresponding training ratio.

of the compared methods do not report standard deviations in this article, and we only used OA. The methods we compared are mainly divided into three categories: methods based on rhe CNN, methods based on the ViT, and methods related to distillation.

1) NWPU Dataset: The results of CASD in comparison to the state-of-the-art and baseline methods are presented in Table II. In this table, we have selected CNN-based models, ViT-based models, and distillation-based models. The accuracy of the CNN-based methods has already reached exceptional results with more than 90% accuracy. For instance, Xu's method exhibits an accuracy of 91.91% and 94.43% under 10% and 20% training samples, respectively. As indicated in Table II, the upper performance limit of the ViT-based method surpasses that of the CNN-based method, which can be attributed to the network structure of the ViT. The benchmark performance of the original ViT was tested under the same data enhancement and optimization strategy as CASD. The results showed that the overall classification accuracy of the ViT reached 92.15% and 94.19% with 10% and 20% training samples, respectively. SCViT, which utilizes both spectral and spatial information, further enhances the classification performance of the NWPU dataset. ET-GSNet, classified as a distillation-based method, leverages the long-range information from the ViT and distills it into ResNet, resulting in improved performance and efficiency. The overall classification accuracy of ET-GSNet was 92.72% and 94.50% with 10% and 20% training samples, respectively.

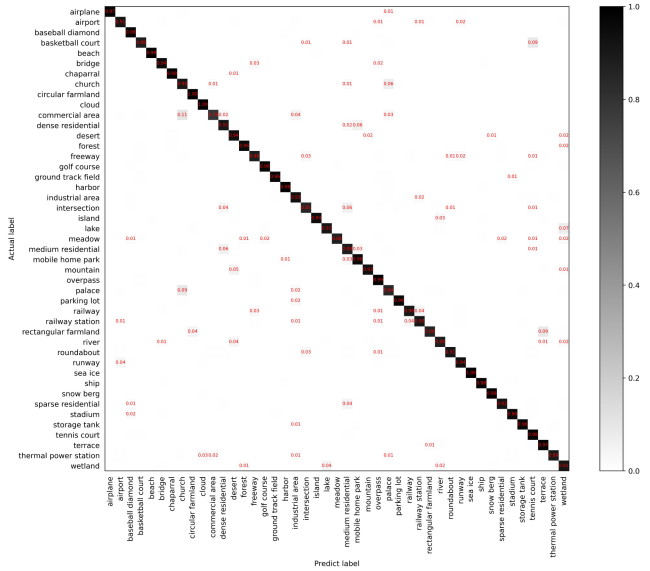


Fig. 9. Confusion matrix is plotted using the CASD-ViT model trained on 10% of the NWPU dataset as the training set.

EMSCNet is a distillation-based multisample contrastive network that exhibits good performance with 10% and 20% training samples. Our proposed method, CASD, demonstrates superior classification accuracy compared to the other methods. Specifically, when equipped with ResNet-50 and ViT, CASD achieved a 1.52% and 0.97% improvement, respectively, under 10% training samples, and a 1.64% and 1.33% improvement, respectively, under 20% training samples. This demonstrates that our method is able to effectively differentiate the similarities and differences among remote sensing images and fully exploit the global spatial relationships of the images.

To further evaluate the performance of CASD-ViT under 10% training samples, we plotted its confusion matrix, as shown in Fig. 9. The results reveal that the correct rate for each category is satisfactory. For instance, categories with high correctness, such as "storage tank" and "harbor," achieved 100% correct rates. This indicates that CASD is able to distinguish different scene contents well and effectively extract higher order semantic information for scenes with complex spatial relationships. On the other hand, categories with high error rates include "church" and "medium residential," with a 90% and 86% correct rate, respectively. These errors occur due to the similarities in architectural style between "church" and "palace" and the ambiguity in the definition of "residential," making it difficult for the network to distinguish between "medium residential" and "dense residential."

2) AID Dataset: As shown in Table III, the overall classification accuracy of the AID dataset has reached a satisfactory level for most of the state-of-the-art methods. The ViT-based approach exhibits superior performance compared to the CNN-based method, reaching a new high performance standard. Under 20% training samples, the baseline models ViT and ResNet-50 achieved accuracies of 94.81% and 94.16%, respectively. With 50% training samples, they achieved accuracies of 96.46% and 95.64% respectively. Meanwhile, the methods SCViT and

TABLE III
ACCURACY (OA ± STD) OF CASD COMPARED WITH OTHER METHODS ON THE AID DATASET

Method	Year	20% train ratio	50% train ratio
CNN-based methods			
ResNet-50 [44]	2015	94.16±0.22	95.64±0.18
VGGNet [42]	2017	86.59±0.29	89.64±0.36
GoogleNet [42]	2017	83.44±0.40	86.39±0.55
ARCNet-VGG16 [17]	2018	88.75±0.40	93.10±0.55
EfficientNet-B0-aux [45]	2019	93.96±0.11	–
EfficientNet-B3-aux [45]	2019	94.19±0.15	–
GBNet [56]	2019	90.16±0.24	93.72±0.34
GBNet+global feature [56]	2019	92.20±0.23	95.48±0.12
Inception-v3-CapsNet [18]	2019	93.79±0.13	96.32±0.12
BiMobileNet [48]	2020	94.83±0.24	96.87±0.23
ACNet [52]	2021	93.33±0.29	95.38±0.29
GLDBS [51]	2021	95.45±0.19	97.01±0.22
EFPN-DSE-TDFF [57]	2021	94.02±0.21	94.50±0.30
Xu's method [22]	2021	94.74±0.23	97.65±0.25
ViT-based Methods			
ViT [28]	2020	94.81±0.12	96.46±0.10
V16_21K[224×224] [29]	2021	94.97±0.01	–
SCViT [33]	2022	95.56±0.17	96.98±0.16
Distillation-based Methods			
ET-GSNet [53]	2022	95.58±0.18	96.88±0.19
DKD [58]	2022	95.09	96.94
ViT-CL [54]	2022	95.60	97.42
EMSCNet-ResNet50 [55]	2023	95.13±0.10	96.96±0.10
Our Methods			
CASD-ResNet50	2023	95.72±0.13	96.96±0.16
CASD-ViT	2023	**96.18±0.20**	**97.64±0.11**

The bold values represent the highest classification accuracy at the corresponding training ratio.

ET-GSNet maintain their high performance, achieving 95.56% and 95.58% classification accuracy under 10% training samples, respectively. Our method, CASD, demonstrates superiority over other methods in terms of classification accuracy. With 20% of training samples, CASD resulted in a 1.56% and 1.37% improvement when integrated into ResNet-50 and ViT, respectively. Similarly, with 50% of training samples, CASD showed improvement of 1.32% and 1.18% when integrated into ResNet-50 and ViT, respectively. The reduction of intraclass variances and increase in interclass distances by CASD play a crucial role in enhancing the performance. The confusion matrix of CASD-ViT under the 20% training sample of the AID dataset is displayed in Fig. 10. The accuracy across all 30 categories is considered to be satisfactory, even with the limited amount of training data. Specifically, scenes such as "baseball field," "beach," and "mountain" achieved a 100% correct rate. On the other hand, some categories, including "center" (0.87), "resort" (0.89), and "square" (0.83), have a correct rate below 90%.

3) UCM Dataset: The performance of CASD is evaluated on the UCM dataset, which has a smaller number of images and categories, to demonstrate its robustness in small-scale datasets. The state-of-the-art methods have achieved high accuracy on this dataset, reaching performance saturation. Table IV shows how

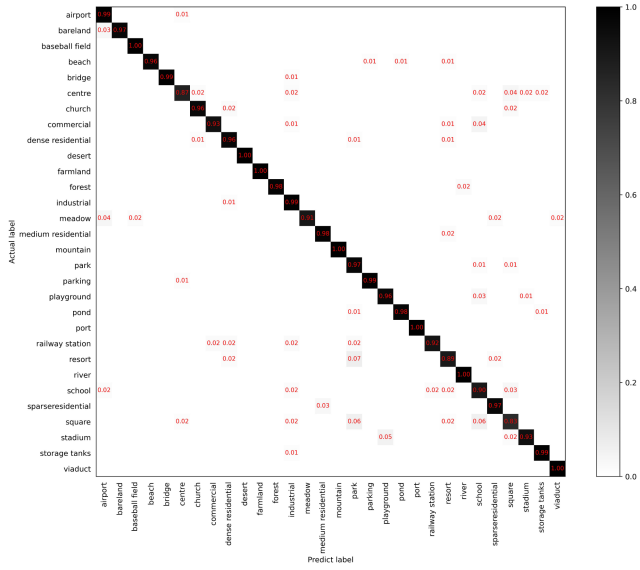


Fig. 10. Confusion matrix is plotted using the CASD-ViT model trained on 20% of the AID dataset as the training set.

TABLE IV
ACCURACY (OA ± STD) OF CASD COMPARED WITH OTHER METHODS ON THE UCM DATASET

Method	Year	50% train ratio	80% train ratio
CNN-based methods			
ResNet-50 [44]	2015	96.88±0.21	98.23±0.13
VGGNet [42]	2017	94.14±0.69	95.21±1.20
GoogleNet [42]	2017	92.70±0.60	94.31±0.89
APDCNet [59]	2019	95.01±0.43	97.05±0.43
SRSCNN [60]	2018	97.88±0.31	98.13±0.33
EfficientNet-B0-aux [45]	2019	98.01±0.45	–
EfficientNet-B3-aux [45]	2019	98.22±0.49	–
VGG-16-CapsNet [18]	2019	98.81±0.22	95.33±0.18
Inception-v3-CapsNet [18]	2019	97.59±0.16	99.05±0.24
Contourlet CNN [46]	2020	–	98.97±0.21
BiMobileNet [48]	2020	98.45±0.27	99.03±0.28
ACNet [52]	2021	–	**99.76±0.10**
EFPN-DSE-TDFF [57]	2021	96.19±0.13	99.14±0.22
Xu's method [22]	2021	98.61±0.22	98.97±0.31
ViT-based Methods			
ViT [28]	2020	97.93±0.24	98.63±0.11
V16_21K[224×224] [29]	2019	98.14±0.47	–
SCViT [33]	2022	98.90±0.19	99.57±0.31
ET-GSNet [53]	2022	–	99.29±0.34
Our Methods			
CASD-ResNet50	2023	98.58±0.13	99.62±0.16
CASD-ViT	2023	**99.07±0.09**	99.70±0.11

The bold values represent the highest classification accuracy at the corresponding training ratio.

CASD compares to other methods in terms of performance. Most methods achieved accuracy above 98% under a 50% training ratio. The baseline model ViT, for instance, achieved an overall classification accuracy of 97.93% and 98.63% under a 50% and 80% training ratio, respectively. Under the 50% training samples, CASD outperformed the other methods, with a 1.70% and

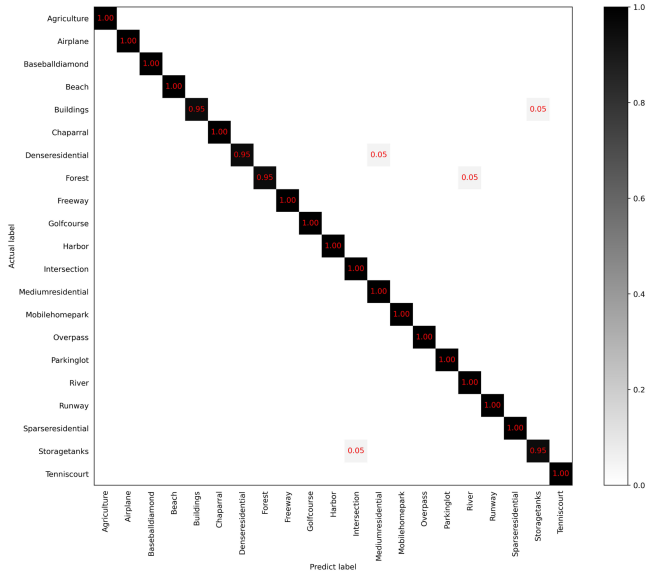


Fig. 11. Confusion matrix is plotted using the CASD-ViT model trained on 50% of the UCM dataset as the training set.

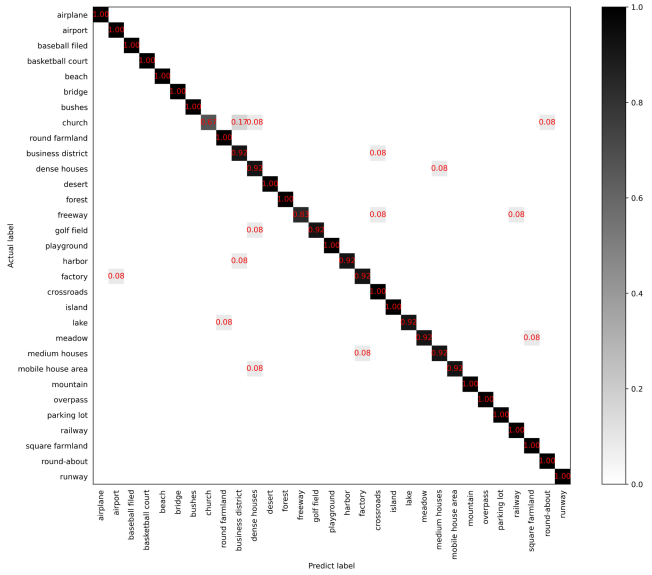


Fig. 12. Confusion matrix is plotted using the CASD-ViT model trained on 80% of the OPTIMAIL-31 dataset as the training set.

TABLE V
ACCURACY (OA ± STD) OF CASD COMPARED WITH OTHER METHODS ON THE OPTIMAL-31 DATASET

Method	Year	80% train ratio
CNN-based methods		
VGGNet [61]	2014	88.40±0.20
ResNet-50 [44]	2015	93.96±0.14
GoogLeNet [62]	2017	84.67±0.25
ARCNet-VGGNet16 [17]	2018	92.70±0.35
ARCNet-Alexnet [17]	2018	85.75±0.35
ARCNet-ResNet [17]	2018	91.28±0.45
KFBNet-VGG16 [63]	2020	95.12±0.13
KFBNet-DenseNet121 [63]	2021	95.60±0.63
ViT-based Methods		
ViT [28]	2020	94.72±0.22
Our Methods		
CASD-ResNet50	2023	95.48±0.23
CASD-ViT	2023	**96.05±0.26**

The bold values represent the highest classification accuracy at the corresponding training ratio.

1.14% improvement when equipped in ResNet-50 and ViT, respectively. Meanwhile, under the 80% training samples, CASD obtained a 1.39% and 1.07% improvement when equipped in ResNet-50 and ViT, respectively. These results, as shown in Table IV, indicate that CASD still has good robustness in achieving better classification performance even in small-scale datasets. Fig. 11 presents the confusion matrix of the CASD-ViT under a 50% training ratio. Out of the 21 categories, 17 achieved a 100% correct rate. The misclassification between the categories "dense residential" and "medium residential" is a difficult issue to avoid.

4) OPTIMAL-31 Dataset: Table V presents the results of comparison between CASD and other methods. The baseline models, such as VGGNet, ResNet-50, GoogLeNet, and ViT, exhibit classification accuracy values of 88.40%, 93.96%, 84.67%, and 94.72%, respectively. CASD was applied to both ResNet-50 and ViT. Under an 80% training sample, CASD-ResNet50 and CASD-ViT achieved classification accuracy values of 95.48% and 96.05%, respectively. In comparison with other methods, such as KFBNet and ARCNetet, CASD demonstrates superior performance and generalization. Furthermore, the confusion matrix of the CASD-ViT is depicted in Fig. 12.

TABLE VI
OVERALL CLASSIFICATION ACCURACY USING CASD ON 1% TRAINING DATA

Method	Acc. (1% NWPU)	Acc. (1% AID)
CASD-ViT	80.54	74.23
CASD-ResNet50	75.11	70.68

To verify the learning capability of CASD under small sample conditions, we conducted experiments using only 1% of the training data. Table VI presents the overall classification accuracy of CASD with this 1% training sample. As can be seen from the table, the CASD-ViT performs remarkably well on the NWPU dataset. Even with only 1% of the training data, the model's accuracy still reaches 80.54%. This indicates that our model can achieve commendable performance even in small sample learning scenarios.

D. Ablation Study

The ablation experiment dismantled the CASD framework to verify the effectiveness of the CASD framework.

1) Structure Ablation: As depicted in Fig. 13, we conducted ablation experiments by disassembling the CASD framework in order to demonstrate its effectiveness. The experiments used the NWPU dataset with 10% training samples and the AID

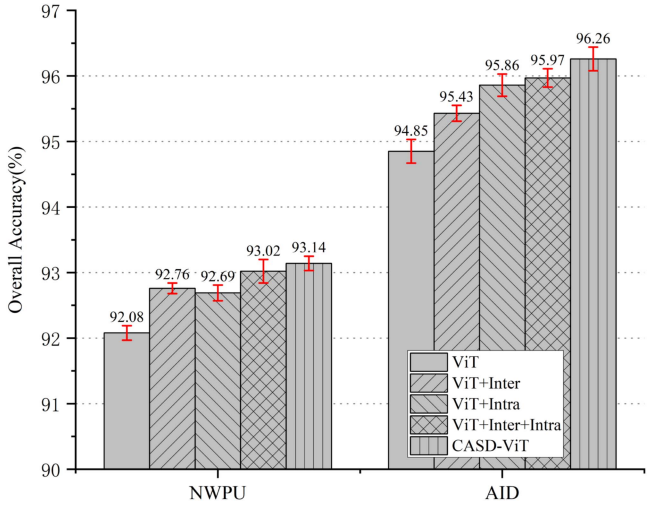


Fig. 13. Ablation studies. OA of different models under the training ratios of 10% on NWPU dataset and 20% on AID dataset. Experimental results are from the validation set.

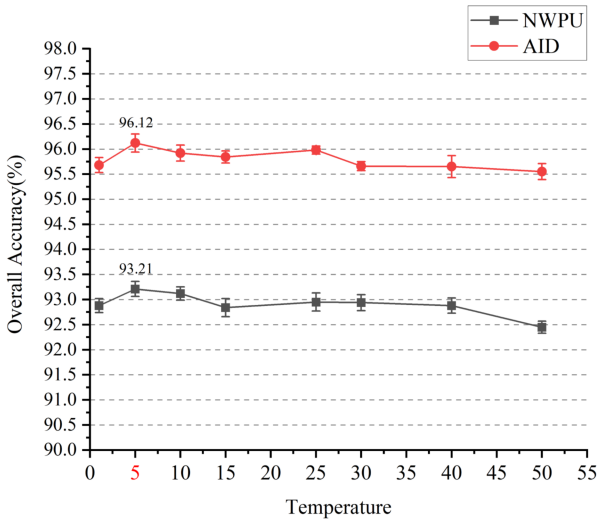


Fig. 14. OAs with different temperature T. In total, 10% training samples of the NWPU dataset and 20% training samples of the AID dataset were used. Experimental results are from the validation set.

dataset with 20% training samples, respectively The ablation experiment is designed as follows.

1) *ViT*: Baseline model.
2) *ViT+inter*: Only distill the knowledge within the class.
3) *ViT+intra*: Only distill the knowledge between classes.
4) *ViT+inter+intra*: Distill the knowledge both within classes and between classes simultaneously.
5) *CASD-ViT*: Our proposed method introduces a learnable interval while distilling the knowledge both within classes and between classes.

As depicted in Fig. 13, the baseline model ViT achieved accuracy scores of 92.08% and 94.85% on the NWPU and AID datasets, respectively. The ViT+inter variant only maximizes the interclass distance between the sample distribution in the feature space, with an optimization objective of $\max(\mathrm{KL}(z_1', z_3'))$. Compared to the baseline model, the accuracy of ViT+inter was improved by 0.68% and 0.58% in the NWPU and AID datasets, respectively. This suggests that optimizing the interclass distance is beneficial in enhancing the classification performance. The results of the experiments on ViT+intra show that optimizing the intraclass distance in the feature space through the optimization objective $\max(\mathrm{KL}(z_1', z_2'))$ can effectively improve the classification performance. The accuracy of ViT+intra is improved by 0.61% and 1.01% compared to the baseline model ViT, on the NWPU and AID datasets, respectively. This demonstrates that optimizing the intraclass distance can effectively improve the classification performance. The ViT+inter+intra framework optimizes both the intraclass and interclass distances in the feature space with the optimization objective defined in (11) by removing α. The experimental results show that the classification accuracy of ViT+inter+intra is improved by 0.26% compared to ViT+inter on the NWPU dataset, and by 0.54% compared to ViT+inter on the AID dataset. These results demonstrate that optimizing both the intraclass and interclass distances can effectively improve the classification performance. The proposed method, CASD-ViT, has an optimization objective as shown in (11). The experimental results show that CASD-ViT achieves a classification accuracy of 93.14% and 96.26% on the NWPU and AID datasets, respectively. The optimization of CASD-ViT involves both the intraclass distance and interclass distance, with the inclusion of the interval α to maintain a larger interclass distance compared to the intraclass distance. The results of the ablation experiments demonstrate the effectiveness of the CASD framework.

2) Distillation Temperature: As demonstrated in Fig. 14, we conducted a search for the optimal distillation temperature ranging from 1 to 50, using 10% training samples from the NWPU dataset and 20% training samples from the AID dataset. As shown in Fig. 14, the best performance of both the NWPU and AID datasets are centered around 5. As a result, the distillation temperature T was selected as 5.

3) Probability Distribution Metric: Our proposed CASD framework uses the KL divergence as a measure of the difference between the two distributions to facilitate the transfer of knowledge. However, alternative measures such as Euclidean distance and Cosine similarity are also viable options. The concept of KL divergence is expressed by (10). The Euclidean distance is a straightforward method of calculating the distance between two vectors. For a pair of vectors A = (a_1, a_2,..., a_n) and B = (b_1, b_2,..., b_n), the calculation for Euclidean distance can be expressed as

$$D(A, B) = \sqrt{(A - B) \cdot (A - B)}. \tag{16}$$

This Euclidean distance quantifies the straight-line distance between the two vectors in an n-dimensional space. On the other hand, to measure the cosine of the angle between vectors A and B, which is an indicator of similarity, we use the following formula:

$$S(A, B) = \frac{A \cdot B}{|A| \cdot |B|}. \tag{17}$$

TABLE VII
PERFORMANCE OF DIFFERENT METRICS ON OPTIMAL-31 DATASET (OA ± STD)

Metric Method \ Our Model	CASD-ViT	CASD-ResNet50
Euclidean Distance	95.89±0.12	94.91±0.17
Cosine Similarity	96.05±0.14	95.22±0.11
KL Divergence	**96.12±0.07**	**95.51±0.10**

The bold values indicate the highest classification accuracy of the same method under three different distribution difference computation metrics.

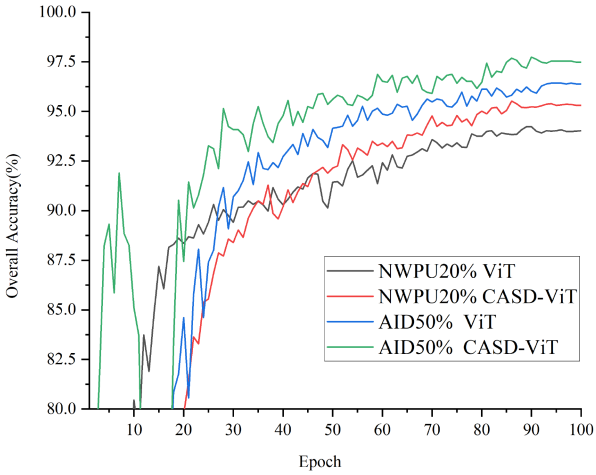


Fig. 15. Training curves of ViT and CASD-ViT. Experiments were conducted using the NWPU dataset under 20% training samples and the AID dataset under 50% training samples.

Here, $S(A, B)$ gives the cosine similarity, a value between -1 and 1, where 1 means the vectors are identical, 0 indicates orthogonality (no correlation), and -1 implies opposite vectors.

As shown in Table VII, we evaluated the accuracy of the proposed method using various metrics. Experiments were carried out on the OPTIMAL-31 dataset utilizing the CASD-ViT and CASD-ResNet50 frameworks. The experimental results indicate that KL divergence, as a metric of probability distribution difference, attains the highest classification performance. Therefore, we adopted KL divergence to measure the intraclass and interclass distances in the distillation process.

E. Visualization Experiment

1) Training Curve Visualization: We visualized the training curves of ViT and CASD-ViT to observe the training process of CASD. We utilized 20% training samples from the NWPU dataset and 50% training samples from the AID dataset. The entire training process was conducted for a duration of 100 epochs, as depicted in Fig. 15. The training process of the ViT exhibits larger fluctuations compared to that of a CNN, particularly in the initial 15 epochs, where the added warm-up strategy causes the learning rate to vary significantly. On the NWPU dataset, the accuracy of the CASD-ViT was not as high in the early stages as that of the ViT, and we surmise that the primary reason for this is the substantial distillation loss in the early stages, which results in a substantial gradient change. However, as the training progressed, the accuracy of the CASD-ViT surpassed that of the ViT, indicating that the model gradually assimilated class knowledge, thus improving its classification performance. On the AID dataset, the CASD-ViT maintained a higher accuracy compared to the ViT until convergence. To conclude, CASD can effectively aid the model in converging to the minima point with greater ease.

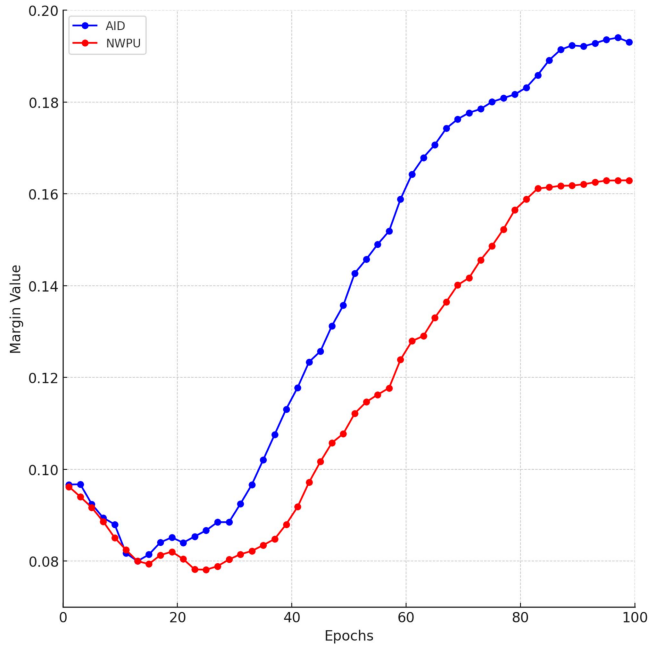


Fig. 16. Learning trend of interval α in CASD-ViT on NWPU and AID datasets.

2) Learnable Interval Visualization: To investigate the learning trend of the learnable interval α within the model, we present the related learning curve. Generally speaking, given an initial value for α, as the network training progresses, will update its value to find an appropriate value as the lower limit between the distance of similar and dissimilar pairs. In Fig. 16, we set the initial value of α to 0.1. As training progresses, α increases with the growth of the epoch count, eventually converging to an optimal value. The growth of α is logical because an increase in α results in a larger interclass distance, facilitating better classification performance.

3) Feature Embedding Visualization:: The CASD framework effectively helps the model generate a more meaningful distribution, reducing intraclass variations and increasing interclass distances. As a result, it addresses the high intraclass diversity and interclass similarity issues commonly encountered in remote sensing image classification. As demonstrated in Fig. 17, we utilized t-SNE [64] to visualize the feature embeddings of both ResNet-50 and CASD-ResNet50. Fig. 17(a) showcases the extensive intraclass variation present in the feature space of ResNet-50, resulting in a crossover between classes. On the other hand, Fig. 17(b) displays the reduction of intraclass variation in the feature space of CASD-ResNet50, which results in clearer boundaries between the classes. Observing Fig. 17(c)–(h), we can get similar conclusions.

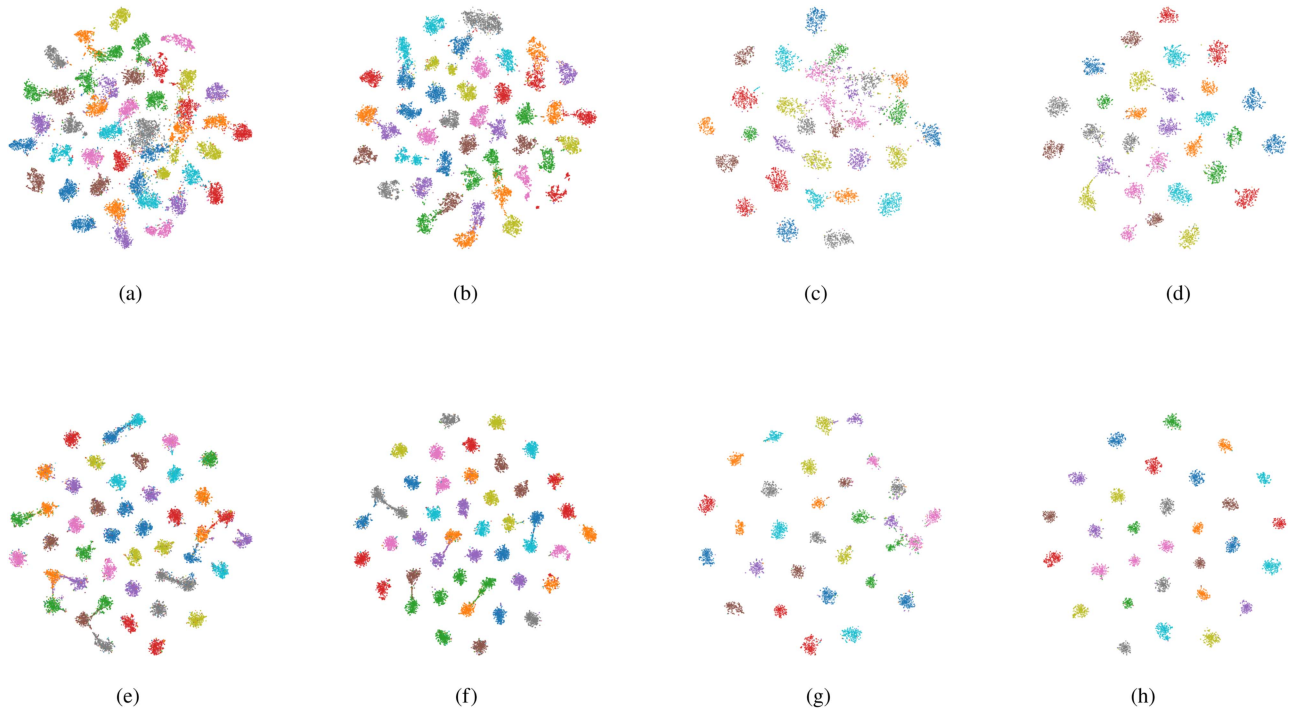


Fig. 17. Visualizing the feature embeddings of baseline and CASD using t-SNE. (a) ResNet-50 NWPU. (b) CASD-ResNet50 NWPU. (c) ResNet-50 AID. (d) CASD-ResNet50 AID. (e) ViT NWPU. (f) CASD-ViT NWPU. (g) ViT AID. (h) CASD-ViT AID.

TABLE VIII
COMPUTATIONAL COMPLEXITY AND EFFICIENCY OF THE MODELS BASED ON THE NWPU DATASET

Model	Parameters	Throughput↑ (image/s)	Acc. (NWPU 10%)
VGGNet [42]	138 M	642	83.59
GooleNet [42]	7 M	1225	81.29
ResNet-50 [44]	24 M	1435	90.69
ViT-S [28]	49 M	910	91.52
ViT-B [28]	86 M	520	92.18
ViT-L [28]	304 M	117	92.75
CASD-ViT	86 M	485	93.15
CASD-ResNet50	25 M	1455	92.16

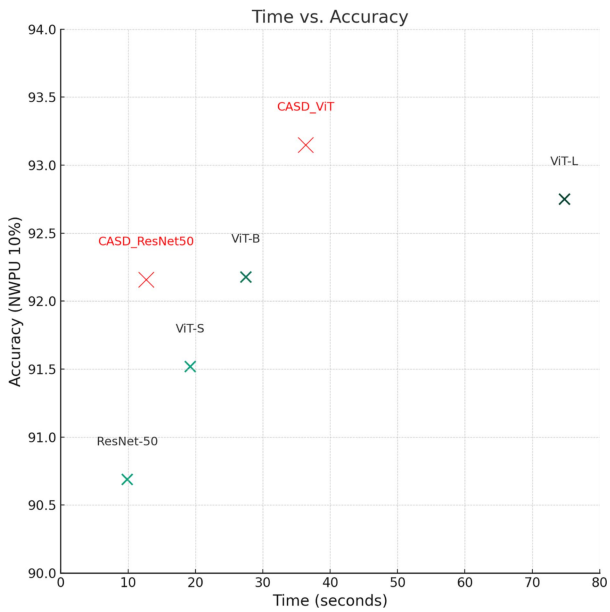


Fig. 18. Visualization of training time and accuracy based on 10% NWPU dataset.

V. DISCUSSION

A. Running Time and Parameters

In the field of deep learning, more complex models often offer higher accuracy, but they might also lead to longer training and inference times. The ideal scenario is to identify a model that strikes a balance between high accuracy and high speed.

To assess the efficiency of our method, we set up two groups of experiments to separately evaluate our approach in both training and inference stages. Compared to the baseline model, our method requires three forward passes and one gradient update during the training phase, resulting in a greater time overhead. However, during the inference stage, our method's efficiency is comparable to that of the baseline model. Fig. 18 illustrates the relationship between time (required to train one epoch) and accuracy during the training phase, while Table VIII depicts the relationship between parameter size, throughput (number of images inferred per second), and classification accuracy in the inference phase.

From Fig. 18, it can be observed that the ViT series models take a significant amount of time to train. This is due to the large number of parameters and the complexity of the ViT models, yet they deliver commendable performance.

Relative to the baseline model, our method does increase the training time but brings about a significant boost in performance. Notably, for CASD-ResNet50, a slight increase in training time

results in a considerable performance gain, making it suitable for practical applications. Furthermore, the training time for the CASD-ViT is much less than that for the ViT-L, but its performance surpasses the ViT-L. This suggests that the CASD-ViT strikes an excellent balance between the training speed and accuracy.

Looking at Table VIII, during the inference stage, our approach is virtually indistinguishable from the baseline model, as we only need to utilize the student branch for inference. Although GoogleNet boasts a higher throughput, its performance is evidently not up to the mark. CASD-ResNet50, while retaining the speed advantages of the baseline ResNet-50 model, significantly enhances the classification accuracy. CASD-ViT achieves the best classification accuracy, surpassing the baseline ViT model. In conclusion, the balance our method achieves between accuracy and speed is commendable.

B. Extended Interpretation of the Model

From the perspective of information entropy, our method can be clearly explained. In simple terms, information entropy measures the uncertainty of a random variable. For a specific probability distribution P, its information entropy $H(P)$ is defined as $\sum p(x) \log p(x)$, where $p(x)$ represents the probability of the random variable taking a certain value.

In KD, softening logits is a common approach. Its aim is to increase the information entropy of the model's output, which can be seen as adding some "noise" to the model. In our method, two teacher branches pass distilled information to the student branch. This means the information received by the student branch has high entropy or, in other words, contains this added uncertainty or "noise." This "noise" is actually precious. It not only allows the student model to learn the original hard label knowledge but also helps it absorb extra knowledge from similar and different samples. This deepens the model's understanding of the differences between various remote sensing image categories, improving the model's robustness and overall performance.

Clustering is an unsupervised learning method, aiming to group similar data points together, forming clear and separate clusters. Spectral clustering is a special variant of clustering methods, exploring the graphical representation of data to discover inherent group structures. In spectral clustering, data points are viewed as nodes in a graph, and a graph is constructed by calculating the similarity between nodes. Then, by analyzing the spectral characteristics of the graph (such as eigenvalues and eigenvectors), clusters are identified and formed.

In the field of remote sensing image interpretation, some related studies also focus on clustering and optimization of feature space. For instance, Doulamis et al. [65] employed constraint inductive learning and spectral clustering methods to support personalized 3-D navigation. Zhang et al. [66] discussed a spectral-spatial sparse subspace clustering method, addressing land-cover classification problems in hyperspectral remote sensing images, while Bach et al. [67] proposed a new objective and algorithm for learning spectral clustering to optimize clustering results.

In contrast to the aforementioned works, our method is not a traditional clustering algorithm, but it optimizes clustering boundaries. To achieve this, we created similar pairs and dissimilar pairs, reducing the feature distance within the same category and increasing the feature distance across different categories. This approach, starting from a clustering perspective, optimizes the distribution in feature space, effectively alleviating the issues of high intraclass diversity and interclass similarity in remote sensing image classification. By reducing intraclass variations and increasing interclass distances, our method provides clearer and more compact clustering results for remote sensing images, thus improving the classification performance.

VI. Conclusion

A CASD framework is proposed for the high intraclass variations and interclass similarity that exist in remotely sensed imagery. This framework utilizes self-distillation to extract knowledge from both same-class and different-class samples, and subsequently, guide the learning process of student models. In detail, the CASD framework operates in the feature space by pushing samples of the same class to be more similar and samples of different classes to be more dissimilar. To further encourage the model to learn distinctive features, we introduce an learnable interval α that makes the interclass distance significantly larger than the intraclass distance. Experiments were performed using two different network architectures, ViT and ResNet-50, and the results showed that the CASD framework has excellent generalization and robustness. Furthermore, our experiments on four publicly available datasets showed that the CASD framework outperforms the state-of-the-art method in terms of classification performance. In future work, we plan to extend the CASD framework to a wider range of remote sensing image classification models.

References

[1] G. Cheng, L. Guo, T. Zhao, J. Han, H. Li, and J. Fang, "Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA," *Int. J. Remote Sens.*, vol. 34, no. 1, pp. 45–59, 2013.

[2] T. Martha, N. Kerle, C. J. van Westen, V. Jetten, and K. V. Kumar, "Segment optimization and data-driven thresholding for knowledge-based landslide detection by object-based image analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4928–4943, Dec. 2011.

[3] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[4] S. Bhagavathy and B. S. Manjunath, "Modeling and detection of geospatial objects using texture motifs," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3706–3715, Dec. 2006.

[5] X. Yao et al., "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[6] S. Cui, "Comparison of approximation methods to Kullback-Leibler divergence between Gaussian mixture models for satellite image retrieval," *Remote Sens. Lett.*, vol. 7, no. 7, pp. 651–660, 2016.

[7] J. Feng, Z. Gao, R. Shang, X. Zhang, and L. Jiao, "Multi-complementary generative adversarial networks with contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Aug. 2023, Art. no. 5520018.

[8] J. Feng, G. Bai, D. Li, X. Zhang, R. Shang, and L. Jiao, "MR-Selection: A meta-reinforcement learning approach for zero-shot hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Dec. 2023, Art. no. 5500320.

[9] Y. Yi and S. Newsam, "Comparing SIFT descriptors and gabor texture features for classification of remote sensed imagery," in *Proc. IEEE 15th Int. Conf. Image Process.*, 2008, pp. 1852–1855.

[10] J. Santos, O. Penatti, and R. Torres, "Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification," in *Proc. 5th Int. Conf. Comput. Vis. Theory Appl.*, 2010, pp. 203–208.

[11] C. Chen, B. Zhang, H. Su, W. Li, and L. Wang, "Land-use scene classification using multi-scale completed local binary patterns," *Signal, Image Video Process.*, vol. 10, no. 4, pp. 745–752, 2016.

[12] O. Penatti, K. Nogueira, and J. Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 44–51.

[13] B. Luo, S. Jiang, and L. Zhang, "Indexing of remote sensing images with different resolutions by multiple features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 4, pp. 1899–1912, Aug. 2013.

[14] Z. Li, Z. Zhou, and D. Hu, "Scene classification using a multi-resolution bag-of-features model," *Pattern Recognit.*, vol. 46, no. 1, pp. 424–433, 2013.

[15] S. d'Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, "ConViT: Improving vision transformers with soft convolutional inductive biases," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2286–2296.

[16] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13876–13885.

[17] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019.

[18] W. Zhang, P. Tang, and L. Zhao, "Remote sensing image scene classification using CNN-CapsNet," *Remote Sens.*, vol. 11, no. 5, pp. 494–515, 2019.

[19] Q. Wang, W. Huang, Z. Xiong, and X. Li, "Looking closer at the scene: Multiscale representation learning for remote sensing image scene classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1414–1428, Apr. 2022.

[20] K. Xu, H. Huang, P. Deng, and Y. Li, "Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5751–5765, Oct. 2022.

[21] Y. Wan, Y. Zhong, A. Ma, J. Wang, and L. Zhang, "E2SCNet: Efficient multiobjective evolutionary automatic search for remote sensing image scene classification network architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, doi: 10.1109/TNNLS.2022.3220699.

[22] C. Xu, G. Zhu, and J. Shu, "A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jan. 2021, Art. no. 5501415.

[23] K. Makantasis, A. Doulamis, N. Doulamis, A. Nikitakis, and A. Voulodimos, "Tensor-based nonlinear classifier for high-order data analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2221–2225.

[24] K. Makantasis, A. D. Doulamis, N. D. Doulamis, and A. Nikitakis, "Tensor-based classification models for hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6884–6898, Dec. 2018.

[25] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[26] Z. Lan et al., "Albert: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[27] A. Vaswani et al., "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 5999–6009, 2017.

[28] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[29] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sens.*, vol. 13, no. 3, pp. 516–534, 2021.

[30] M. Kaselimi, A. Voulodimos, I. Daskalopoulos, N. Doulamis, and A. Doulamis, "A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3299–3307, Jul. 2023.

[31] P. Deng, K. Xu, and H. Huang, "When CNNs meet vision transformer: A joint framework for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Sep. 2021, Art. no. 8020305.

[32] J. Zhang, H. Zhao, and J. Li, "TRS: Transformers for remote sensing scene classification," *Remote Sens.*, vol. 13, no. 20, 2021, Art. no. 4143.

[33] P. Lv, W. Wu, Y. Zhong, F. Du, and L. Zhang, "SCViT: A spatial-channel feature preserving vision transformer for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Mar. 2022, Art. no. 4409512.

[34] S. Hao, B. Wu, K. Zhao, Y. Ye, and W. Wang, "Two-stream swin transformer with differentiable sobel operator for remote sensing image classification," *Remote Sens.*, vol. 14, no. 6, 2022, Art. no. 1507.

[35] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Comput. Sci.*, vol. 14, no. 7, pp. 38–39, 2015.

[36] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3713–3722.

[37] L. Zhang, C. Bao, and K. Ma, "Self-distillation: Towards efficient and compact neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4388–4403, Aug. 2022.

[38] D. Wang, J. Zhang, B. Du, G.-S. Xia, and D. Tao, "An empirical study of remote sensing pretraining," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, May 2022, Art. no. 5608020.

[39] Z. Duan, S. Wang, H. Di, and J. Deng, "Distillation remote sensing object counting via multi-scale context feature aggregation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Nov. 2021, Art. no. 5613012.

[40] Y. Hu, X. Huang, X. Luo, J. Han, X. Cao, and J. Zhang, "Variational self-distillation for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Jul. 2022, Art. no. 5627313.

[41] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[42] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[43] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[45] Y. Bazi, M. M. Al Rahhal, H. Alhichri, and N. Alajlan, "Simple yet effective fine-tuning of deep CNNs using an auxiliary classification loss for remote sensing scene classification," *Remote Sens.*, vol. 11, no. 24, 2019, Art. no. 2908.

[46] M. Liu, L. Jiao, X. Liu, L. Li, F. Liu, and S. Yang, "C-CNN: Contourlet convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2636–2649, Jun. 2021.

[47] Z. Zhao, Z. Luo, J. Li, C. Chen, and Y. Piao, "When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework," *Remote Sens.*, vol. 12, no. 20, 2020, Art. no. 3276.

[48] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, "An efficient and lightweight convolutional neural network for remote sensing image scene classification," *Sensors*, vol. 20, no. 7, 2020, Art. no. 1999.

[49] Q. Bi, H. Zhang, and K. Qin, "Multi-scale stacking attention pooling for remote sensing scene classification," *Neurocomputing*, vol. 436, pp. 147–161, 2021.

[50] X. Wang, L. Yuan, H. Xu, and X. Wen, "CSDS: End-to-end aerial scenes classification with depthwise separable convolution and an attention mechanism," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10484–10499, Oct. 2021.

[51] K. Xu, H. Huang, and P. Deng, "Remote sensing image scene classification based on global–local dual-branch structure model," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, May 2021, Art. no. 8011605.

[52] X. Tang, Q. Ma, X. Zhang, F. Liu, J. Ma, and L. Jiao, "Attention consistent network for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2030–2045, Jan. 2021.

[53] K. Xu, P. Deng, and H. Huang, "Vision transformer: An excellent teacher for guiding small networks in remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Feb. 2022, Art. no. 5618715.

[54] M. Bi, M. Wang, Z. Li, and D. Hong, "Vision transformer with contrastive learning for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 738–749, Dec. 2023.

[55] Y. Zhao, J. Liu, J. Yang, and Z. Wu, "EMSCNet: Efficient multisample contrastive network for remote sensing image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Mar. 2023, Art. no. 5605814.

[56] H. Sun, S. Li, X. Zheng, and X. Lu, "Remote sensing scene classification by gated bidirectional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 82–96, Jan. 2020.

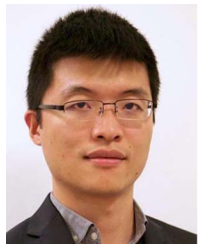
[57] X. Wang, S. Wang, C. Ning, and H. Zhou, "Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7918–7932, Sep. 2021.
[58] D. Li, Y. Nan, and Y. Liu, "Remote sensing image scene classification model based on dual knowledge distillation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Sep. 2022, Art. no. 4514305.
[59] Q. Bi, K. Qin, H. Zhang, J. Xie, Z. Li, and K. Xu, "APDC-Net: Attention pooling-based convolutional network for aerial scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 9, pp. 1603–1607, Sep. 2020.
[60] Y. Liu, Y. Zhong, F. Fei, Q. Zhu, and Q. Qin, "Scene classification based on a deep random-scale stretched convolutional neural network," *Remote Sens.*, vol. 10, no. 3, pp. 444–466, 2018.
[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[62] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[63] F. Li, R. Feng, W. Han, and L. Wang, "High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8077–8092, Nov. 2020.
[64] L. Van der Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
[65] N. Doulamis, C. Yiakoumettis, G. Miaoulis, and E. Protopapadakis, "A constraint inductive learning-spectral clustering methodology for personalized 3 d navigation," in *Proc. Int. Symp. Vis. Comput.*, 2013, pp. 108–117.
[66] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.
[67] F. Bach and M. Jordan, "Learning spectral clustering," *Adv. Neural Inf. Process. Syst.*, vol. 16, pp. 305–312, 2003.

Bin Wu received the B.S. degree in automation from the Weifang Institute of Technology, Weifang, China, in 2020. He is currently working toward the M.S. degree in electronic information with the Qingdao University of Technology, Qingdao, China.

His research interests include computer vision and remote sensing image processing.

Siyuan Hao (Member, IEEE) received the Ph.D. degree in information and communication engineering from the College of Information and Communications Engineering, Harbin Engineering University, Harbin, China, in 2015.

She is currently an Associate Professor with Beijing Jiaotong University, Beijing, China, where she teaches remote sensing and electrical communication. Her research interests include hyperspectral imagery processing and machine learning.



Wei Wang received the Ph.D. degree in information and communication from the University of Trento, Trento, Italy, in 2018.

He is currently a Professor with the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China. Before joining BJTU, he was an Assistant Professor with the University of Trento. His research interests include computer vision, deep learning, and augmented reality, particularly human-centered perception, including face and hand analysis.