



OPEN Pure data correction enhancing remote sensing image classification with a lightweight ensemble model

Huaxiang Song[✉], Hanglu Xie, Yingying Duan, Xinyi Xie, Fang Gan, Wei Wang & Jinling Liu

The classification of remote sensing images is inherently challenging due to the complexity, diversity, and sparsity of the data across different image samples. Existing advanced methods often require substantial modifications to model architectures to achieve optimal performance, resulting in complex frameworks that are difficult to adapt. To overcome these limitations, we propose a lightweight ensemble method, enhanced by pure data correction, called the Exceptionally Straightforward Ensemble. This approach eliminates the need for extensive structural modifications to models. A key innovation in our method is the introduction of a novel strategy, quantitative augmentation, implemented through a plug-and-play module. This strategy effectively corrects feature distributions across remote sensing data, significantly improving the performance of Convolutional Neural Networks and Vision Transformers beyond traditional data augmentation techniques. Furthermore, we propose a straightforward algorithm to generate an ensemble network composed of two components, serving as the proposed lightweight classifier. We evaluate our method on three well-known datasets, with results demonstrating that our ensemble models outperform 48 state-of-the-art methods published since 2020, excelling in accuracy, inference speed, and model compactness. Specifically, our models achieve an overall accuracy of up to 96.8%, representing a 1.1% improvement on the challenging NWPU45 dataset. Moreover, the smallest model in our ensemble reduces parameters by up to 90% and inference time by 74%. Notably, our approach significantly enhances the performance of Convolutional Neural Networks and Vision Transformers, even with limited training data, thus alleviating the performance dependence on large-scale datasets. In summary, our data-driven approach offers an efficient, accessible solution for remote sensing image classification, providing an elegant alternative for researchers in geoscience fields who may have limited time or resources for model optimization.

Keywords Exceptionally straightforward ensemble, Convolutional neural network, Vision transformer, Remote sensing image classification, Deep learning

Background

Remote Sensing Images (RSIs) are pivotal photogrammetric records in Earth monitoring. The rapid increase in onboard sensors has led to a vast accumulation of data derived from diverse imaging techniques, rendering traditional manual interpretation inefficient. Consequently, deep learning (DL) algorithms have been extensively employed in RSI recognition across various domains¹. Among these algorithms, classification is deemed fundamental and vital due to its substantial impact on the performance of other algorithms^{2,3}.

Currently, DL-based methodologies are leading the field of RSI classification. Pre-training on large-scale datasets can accelerate the re-training process for subsequent tasks. As a result, most cutting-edge methods utilize pre-trained Convolutional Neural Network (CNN) or Vision Transformer (ViT) models, which were developed on ImageNet-1 K, for RSI recognition. However, most advanced methods typically require a comprehensive reconfiguration of the model's structure, incorporating sub-modules or sub-models to achieve layered accuracy. Specifically, prior studies have indicated that ViTs may not achieve accuracy as remarkable as CNNs, particularly when training samples are limited, as ViTs have significant performance dependence on large-scale training samples. Consequently, previous CNN- or ViT-based methods often lead to significantly expanded model sizes, complex pipelines, and reduced inference speeds⁴. These conditions present challenges for geoscience researchers, who may lack the necessary expertise or time to utilize these techniques. We propose that this predicament can be partially attributed to the issue of pre-training invalidation.

School of Geography Science and Tourism, Hunan University of Arts and Science, Changde 415000, China. ✉email: cn11028719@huas.edu.cn

Method	Roadmap	AID30	
		TR-20%	TR-50%
LS-EfficientNet ⁵	Fine-Tuning Single CNN	96.86 ± 0.07	98.05 ± 0.06
Simple combined strategy	Combining two CNNs	96.28 ± 0.09	97.77 ± 0.01
Simple fusion strategy	Multi-layer feature fusion	88.52 ± 0.23	90.91 ± 0.17

Table 1. Overall accuracy comparison of different strategies on the AID30 dataset (%).

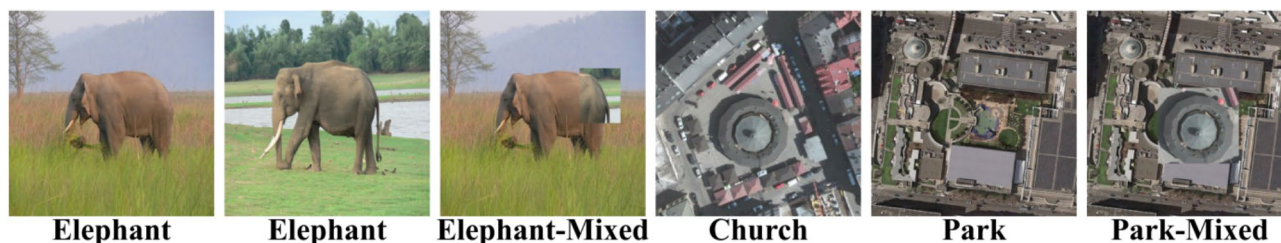


Fig. 1. Differences in background complexity between RSIs and natural images (Source: Authors own work by Python version 3.9).

Pre-training is widely recognized for providing deep models with a stronger starting point in the solution space and addressing statistical challenges when limited data is available for downstream tasks. By pre-training a modified model on large-scale datasets such as ImageNet-1 K, models can leverage prior knowledge, which enhances generalization and overall performance. However, many existing approaches that use modified models often neglect the crucial step of pre-training, resulting in a training process that deviates from standard transfer learning practices. This deviation undermines the effectiveness of fine-tuning, making it less successful in optimizing the modified model. The following empirical evidence supports these concerns, showing that insufficient pre-training significantly hampers the model's performance during fine-tuning.

For instance, LS-EfficientNet⁵ highlighted that pre-trained CNN models often lead to lower accuracy in classification tasks, even when the bottom layers of the pre-trained model are initialized with random weights. Similarly, our study found that combining or fusing features from two CNN models frequently results in inferior performance compared to using an unmodified model with a standard transfer learning approach, as in LS-EfficientNet. This is further corroborated by the results presented in Table 1, where simple strategies such as concatenating the final layers of two EfficientNet-B0 models yield suboptimal accuracy, as indicated by the training ratio (TR) values. These findings underscore the importance of proper pre-training in achieving optimal model performance.

Remote sensing data display a considerable domain discrepancy when compared to natural images. While neural architecture search holds the potential to identify superior models for classifying RSIs, previous approaches have generally focused on relatively small RSI datasets containing tens of thousands of samples^{6–8}, which presents a statistical challenge when compared to large-scale datasets like ImageNet-1 K, which consists of one million samples. Additionally, RSI samples typically feature more complex scenes, with multiple objects and noisier backgrounds, further complicating the classification task. As a result, training augmentation strategies developed for ImageNet-1 K may be less effective when directly applied to RSIs without appropriate modifications. These factors highlight the need for tailored approaches that account for the distinct characteristics of RSIs, rather than simply replicating techniques developed for natural image datasets.

As shown in Fig. 1, the three natural images on the left clearly distinguish between semantic targets and their backgrounds. As a result, regularization techniques, such as CutMix⁹, typically have a low probability of misclassifying a target, such as transforming the 'elephant' category into 'grassland.' In contrast, the three remote sensing samples on the right feature more complex targets and noisier backgrounds. For instance, the cut-and-mixed sample at the rear resembles a church class, even though the majority of the scene corresponds to a park. This illustrates that RSIs often exhibit greater overlap between categories than natural images, which complicates the classification task. Consequently, regularization strategies like CutMix should be applied with reduced strength during training to account for the more challenging characteristics of RSIs.

As illustrated in Fig. 2, the intensity of sunlight plays a crucial role in determining the imaging quality of remote sensing data. Additionally, fluctuations in other imaging conditions, such as direction and blur, can significantly affect the quality of the samples. To ensure the majority of RSIs meet quality standards, we typically use a checking algorithm to filter out low-quality samples. In this process, low-quality samples are represented by red, blue, or black dots, while normal samples are marked in green. A classic RSI dataset in a real-world task is depicted by subset A in Fig. 2, which consists of mostly high-quality samples, with some abnormal ones. However, RSI datasets often contain only around ten thousand samples, leading to a training subset that may suffer from statistical challenges when compared to the full dataset's distribution. For example, subset B in

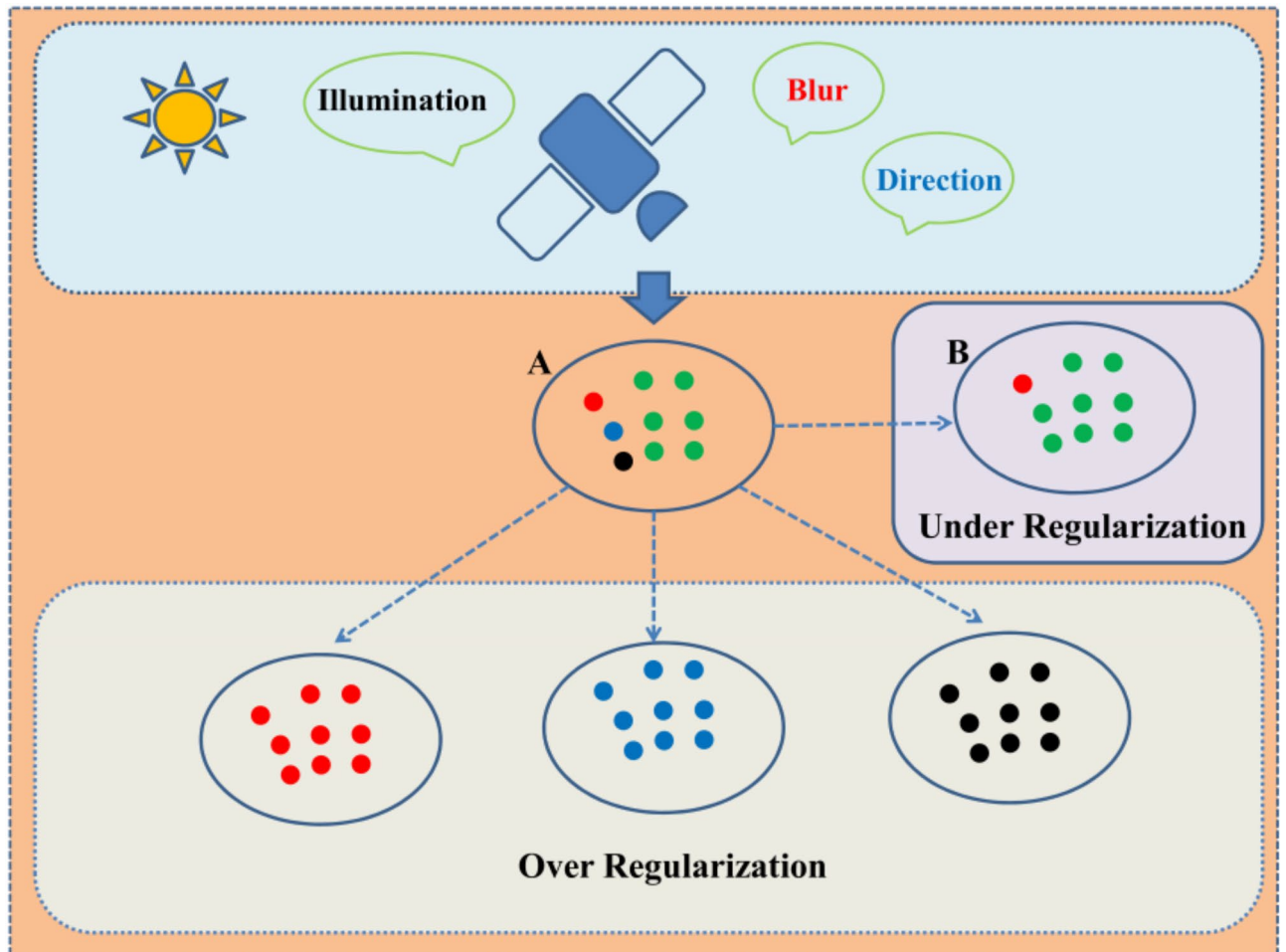


Fig. 2. Challenges in qualitative augmentation (Source: Authors own work by Python version 3.9).

Fig. 2 demonstrates how a randomly selected subset, designed to prevent overfitting, may fail to capture the full variability of the data. To address this under-regularization issue, previous studies typically apply qualitative augmentation strategies. However, as shown at the bottom of Fig. 2, such qualitative pipelines often convert all data points into a single, abnormal type, as the transformation functions remain active throughout. As a result, traditional qualitative strategies frequently lead to biased samples and over-regularization.

Motivation

To overcome the challenges identified in previous approaches, we introduce a novel training augmentation strategy called quantitative augmentation (QA). The primary objective of QA is to address data distribution imbalances within remote sensing data, thereby improving the classification accuracy of CNN and ViT models. In addition to QA, we enhance two existing regularization techniques—CutMix and Online Label Smoothing (OLS)¹⁰—to offer a more adaptable method for developing diverse classifiers. Building on these improvements, we also propose the Exceptionally Straightforward Ensemble (ESE), a method that combines two lightweight models—either a CNN and a ViT or two CNNs—into an ensemble network. The key contributions of our approach can be categorized into three main areas, each of which plays a pivotal role in enhancing model performance and flexibility.

Firstly, the QA and its implementation, the QA Module (QAM), are more effective than traditional data augmentation in RSI classification. The QAM, a plug-and-play unit, is exclusively active during the processing of training samples. As a result, the QA can function in conjunction with any CNN or ViT to enhance performance. Secondly, ESE introduces a more straightforward algorithm to generate more robust ensemble classifiers. It is more convenient compared to traditional, complex methods of generating ensemble models, such as repetitively manipulating training samples or combining too many models. Lastly, our method provides comprehensive algorithms for leveraging open-source models while eliminating the need for extensive experience in re-modifying models. It makes the method both time-efficient and easily reproducible, a notable advantage over existing methods.

We used three benchmark datasets for our evaluation. The results show that our ESE method outperforms 48 advanced techniques introduced between 2020 and 2024, significantly improving accuracy while reducing both the number of parameters and inference time. Additionally, our QA-based CNN and ViT models consistently

exceed the performance of other methods using the same model, even if training samples are limited. We summarize the key contributions of our work as follows:

- (1) We propose a novel augmentation strategy, called QA, along with its implementation, the QAM, to improve the performance of CNNs and ViTs in RSI classification. The QA method effectively aligns the data distribution between training and testing subsets, enabling the models to outperform existing state-of-the-art techniques, particularly in scenarios with limited training samples. As a result, our strategy significantly reduces the reliance on large-scale training datasets, which is a common challenge when applying ViTs to RSI classification tasks.
- (2) We introduce a lightweight ensemble method, termed ESE, to further boost the performance of CNNs or ViTs in the RSI classification. The ESE model, with its compact size and concise generating algorithm, outperforms other advanced methods by a significant margin. For instance, on the most challenging NWPU45 dataset, our ESE models achieved an OA of up to 96.8%, approximately 1.1% higher than the top-ranked method in existing literature. Notably, the improvement within our smallest ESE is accompanied by a reduction of up to 90% in parameters and 74% in inference time, respectively.
- (3) Our method presents an efficient and straightforward way to boost the performance of existing models. Importantly, it sidesteps the complexities associated with model reconfiguration. As a result, it significantly reduces the barriers to developing DL methods for RSI classification tasks. This accessibility is particularly advantageous for geoscience researchers who may have limited experience with model modifications.

The structure of this paper is as follows: section “[Related works](#)” presents a review of relevant literature. Section “[Methodologies](#)” outlines the methodologies employed in this study, including the QA strategy, QAM, method framework, mathematical foundations, and other key parameters. Section “[Experiments and results](#)” provides a comprehensive analysis of the experimental results. Finally, section “[Conclusions](#)” concludes the paper and summarizes the findings of the study.

Related works

Single model approach

Researchers often rely on transfer learning for RSI classification, using deep models pre-trained on ImageNet-1 K, as seen in models like LS-EfficientNet⁵. However, some studies have explored alternative approaches, such as handcrafted designs or neural architecture searches, to identify more effective solutions distinct from traditional transfer learning. For example, TPENAS⁶ proposes a two-phase evolutionary neural architecture search framework to optimize deep learning models for RSI classification, with the first phase determining the optimal model depth and the second phase searching for the best overall structure. Similarly, LCNN-HWCF¹¹ addresses RSI classification challenges through dimension-wise convolution and hierarchical-wise convolution fusion, reducing model parameters and computational complexity while maintaining a balance between speed and performance. Although these methods offer smaller model sizes than established CNNs pre-trained on ImageNet-1 K, they rely solely on RSI datasets for training and validation, lacking the benefit of prior knowledge from large-scale datasets, which results in comparatively lower performance than transfer learning-based methods like LS-EfficientNet.

With the emergence of ViT, this new architecture has been quickly adopted and proven effective for RSI classification. For example, ET-GSNet¹² uses ViT as a teacher to guide a smaller ResNet18 student model through knowledge distillation. The method improves classification performance without increasing computational complexity by smoothly transferring dark knowledge from the teacher model to the student model. Compared to CNN, a ViT usually requires more training samples to reach convergence. Consequently, RSP-ViT¹³ uses the MillionAID dataset with a million samples to train their models from scratch. The results show that RSP-ViT significantly improves performance but still faces challenges in addressing task discrepancies across different downstream tasks.

Among various single-model approaches, a common strategy involves incorporating self-designed modules into pre-trained models. For instance, EAM-Net¹⁴ introduces an enhanced attention module to improve the feature extraction and generalization capabilities of CNNs, while SCViT¹⁵ combines both spatial and channel-wise information through a multi-head self-attention mechanism, a lightweight channel attention module to prioritize important channels, and a progressive patch aggregation technique to preserve local structural features. However, these methods often train the modified models directly on RSI datasets without first retraining them on large-scale datasets. As a result, despite theoretical differences in architecture that might suggest improved accuracy, the experimental outcomes show that these methods exhibit similar performance. This indicates that the lack of large-scale pretraining hinders their ability to achieve significant gains.

Feature fusion approach

Since 2022, many feature fusion-based methods have been proposed, achieving notable performance improvements. For instance, GCSA-Net¹⁶ introduces a novel approach that combines global context spatial attention with densely connected convolutional networks to effectively capture multiscale global features. Similarly, ACGLNet¹⁷ employs a dual-model deep feature fusion strategy, integrating two CNNs to extract complementary multiscale features, using spatial attention to filter out redundant background information, and applying bilinear fusion to combine global and local features. Additionally, SF-MSFormer¹⁸ presents a spatial-frequency multiscale Transformer framework, which integrates spatial-domain and frequency-domain branches to capture comprehensive global multiscale features. However, a common drawback of these methods is the absence of retraining on large-scale datasets, which limits their competitive performance and generalization across diverse datasets.

Furthermore, researchers have extensively investigated techniques for combining multiple models, demonstrating competitive performance but sacrificing their model size and inference speed. For instance, GRMA-Net¹⁹ introduces a gated recurrent multi-attention neural network that leverages multilevel attention modules to focus on informative regions and capture discriminative features from both shallow and deep layers. It further incorporates a deep-gated recurrent unit to model long-range dependencies and contextual relationships. By comparison, P2FEViT²⁰ introduces a hybrid architecture combining CNNs and ViTs to enhance classification performance by capturing both global context and local multimodal information. This approach leverages CNN features embedded into the ViT framework, improving classification performance, reducing reliance on large-scale pre-training data, and accelerating model convergence with minimal training data.

Local features are critical for pattern recognition in RSI samples, especially when handling low-resolution images. To address the limitations of ViTs in capturing these features, TSTNet²¹ introduces a two-stream Swin Transformer network that combines original image features with edge information through a differentiable edge Sobel operator module in the edge stream. Additionally, IBswin-CR²² enhances the Swin Transformer by incorporating an inductive biased shifted window multi-head self-attention module, a random dense sampler, and cyclic regression loss. These modifications effectively capture long-range dependencies and spatial information in RSI samples. Despite these advancements, a common drawback of ViT-based methods is their large model sizes, which can limit efficiency and scalability.

Ensemble approach

Ensemble-based methods have become increasingly prevalent in satellite data analysis, with applications ranging from hyperspectral image processing^{23,24} to land cover change detection^{25,26}. Since 2020, several ensemble approaches have been proposed for RSI classification. For instance, the CNN-HMM model²⁷ combines a modified multi-scale CNN with a hidden Markov model, effectively extracting multi-scale structural features while minimizing computational complexity. This method leverages CNNs for feature fusion and HMMs for context information mining. In comparison, MGML-FENet²⁸ introduces a multigranularity multilevel feature ensemble network by combining a multigranularity feature fusion branch and a multigranularity feature ensemble module to effectively extract and combine diverse features across different network levels. This approach leverages channel-separate and full-channel feature generators to enhance feature extraction and minimize the impact of confusing information. Similarly, ESD-MBENet²⁹ employs a compact multibranch ensemble network that enhances feature representation by fusing output logits and intermediate feature maps. It addresses the challenges of model complexity and inference efficiency by sharing weights across branches and incorporating feature augmentation. Furthermore, the integration of a self-distillation technique enables the main branch to approximate the performance of the full ensemble, significantly reducing computational overhead during inference. Despite these advancements, a common drawback of these methods is their large model sizes, which can limit efficiency and scalability.

Currently, CNN-Transformer hybrid models have demonstrated the potential to leverage multi-modal advantages^{30,31}. Furthermore, efficient classifiers are crucial for real-world remote sensing applications, such as onboard satellite image classification³² or few-shot satellite image classification^{33,34}. However, to the best of the authors' knowledge, efficient yet accurate methods using a hybrid ensemble approach are quite rare in RSI classification.

Methodologies

Structural components of QAM

The training framework for CNN or ViT models is depicted at the top of Fig. 3. As indicated by the red solid arrows, the original RSI sample is first processed by the QAM. The outputs from the QAM are then fed into the models for training. As shown by the red dotted arrow, the m-OLS updates its soft labels at each training epoch.

As illustrated at the bottom of Fig. 3, the QAM comprises a series of Gated Regularization Operators (GROs) arranged in sequence. Notably, each GRO has a shortcut that outputs the original inputs. Let x_i and x'_i represent the original RSI samples and the outputs of the GROs, respectively. Let P_n denote the threshold value of probability to activate the transformation, and let p_n denote the result of the probability calculator in the GROs. The regularization algorithm in the GROs, denoted as f_{GRO} , can be described as follows:

$$x'_i = \begin{cases} x_i, & p_n < P_n \\ f_{GRO}(x_i), & p_n \geq P_n \end{cases} \quad (1)$$

At present, we utilize a maximum of eight GROs within the QAM. These regularization functions include color jitter, horizontal and vertical flips, rotation, grayscale, auto contrast, Gaussian blur, and CutMix, arranged sequentially. We have chosen these functions only from the PyTorch library, which facilitates the replication of our methods by readers.

Importantly, we set a threshold for the activation of the GROs to ensure a quantitative application of these regularization functions. The thresholds for the grayscale and auto contrast functions are specifically set at 0.3, while those for the remaining functions are set at 0.5 according to the ablation experiment results proposed in subsequent sections. Furthermore, we have deactivated several GROs for certain ViT models due to their inferior bias-inducing capabilities compared to CNNs. Namely, we can dynamically adapt the architecture of the QAM to accommodate different models or datasets.

Architecture of the ESE

The architecture of the ESE is illustrated in Fig. 4. In essence, the ESE architecture consists of two individual classifiers chosen from the five candidate models within the table in Fig. 4, each with fixed weights. These

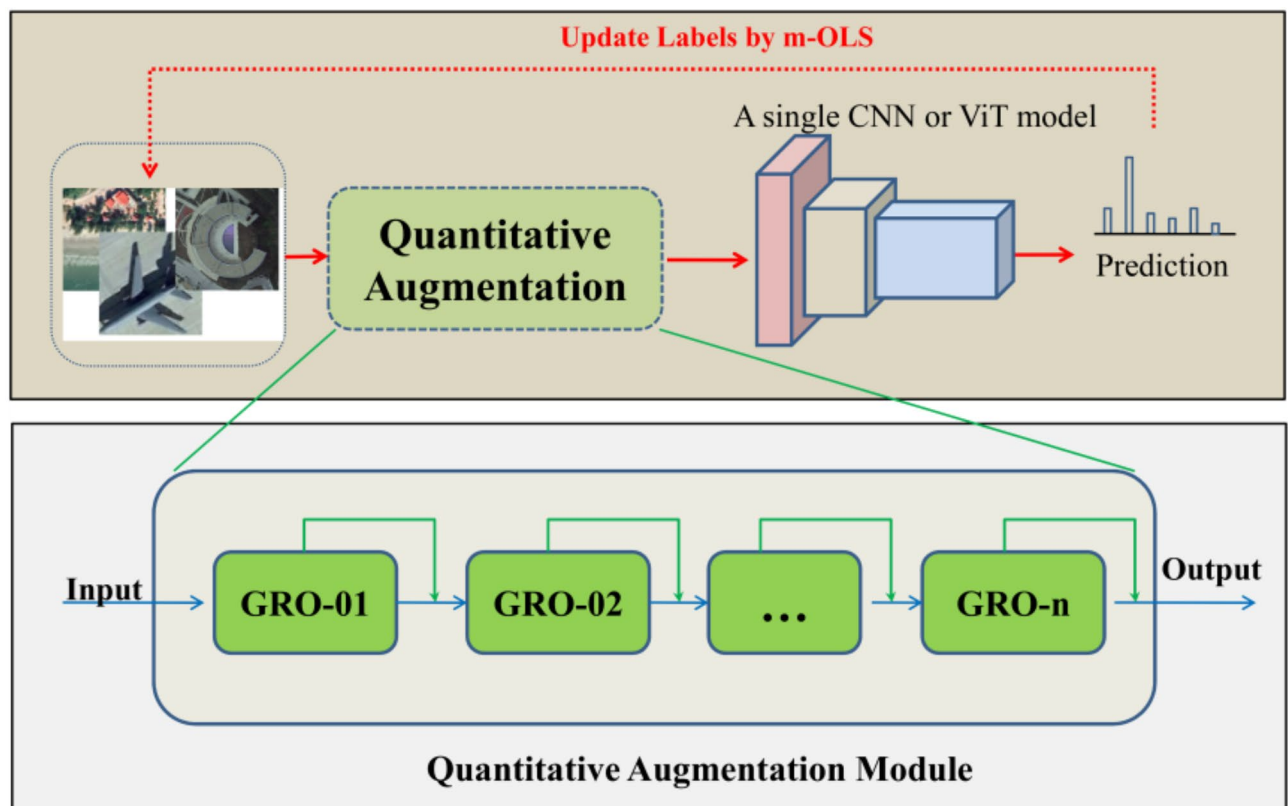


Fig. 3. Training framework and structural components of QAM (Source: Authors own work by Python version 3.9).

classifiers are independently trained on RSI datasets. During inference, the prediction scores from both classifiers are weighted by a hyperparameter, denoted as α . These weighted outputs are subsequently fused to generate the ensemble's final prediction.

Although the accuracy of the ESE can be further enhanced by incorporating additional individual classifiers, experimental results show that performance improvements become marginal as the number of classifiers increases. Therefore, the ESE is configured with two classifiers as the default setting to balance accuracy, model size, and inference speed.

Implementation of the ESE

The ensemble method typically includes bagging, boosting, and stacking³⁵, while ensembles consisting of different classifiers have been widely proven to outperform a single classifier^{36,37}. Generally, bagging and boosting manipulate training sets or training algorithms to obtain diverse classifiers, while stacking uses different models. Accordingly, previous ensemble methods for RSI classification often consist of complex pipelines, too many components, and huge model sizes. In contrast, our ESE method has only two classifiers, C_A and C_B . We first weight the outputs of C_A and C_B using one hyperparameter, then use the sum of the weighted outputs as the ensemble's output.

The benefits of the ESE are threefold: First, the ESE is a straightforward ensemble created by stacking two distinct CNN or ViT models. In comparison to prior ensemble methods, the ESE significantly reduces the number of individual classifiers and parameters, eliminating the need for additional remodeling processes. Second, the training process of its individual classifiers only utilizes traditional fine-tuning, thereby excluding any modifications to the open-source models. Finally, the hybrid ESE straightforwardly leverages the strengths of both CNNs and ViTs, resulting in a more robust yet efficient model. Let F_{ESE} denote the ESE ensemble model. Then, F_{ESE} can be expressed as follows:

$$F_{ESE} = \alpha \times C_A + (1 - \alpha) \times C_B, \quad (2)$$

where α is a hyperparameter set to $[0, 1]$.

In accordance with the definition provided in Eq. (2), we proceed to choose 99 classifiers from the ESE's candidate hypothesis set, utilizing a minimum α change of 0.01. Specifically, the first selected classifier has an α value of 0.01, while the 99th classifier has an α value of 0.99. Following this, we validate these 99 classifiers and select the one with the highest OA as the optimal solution for ESE.

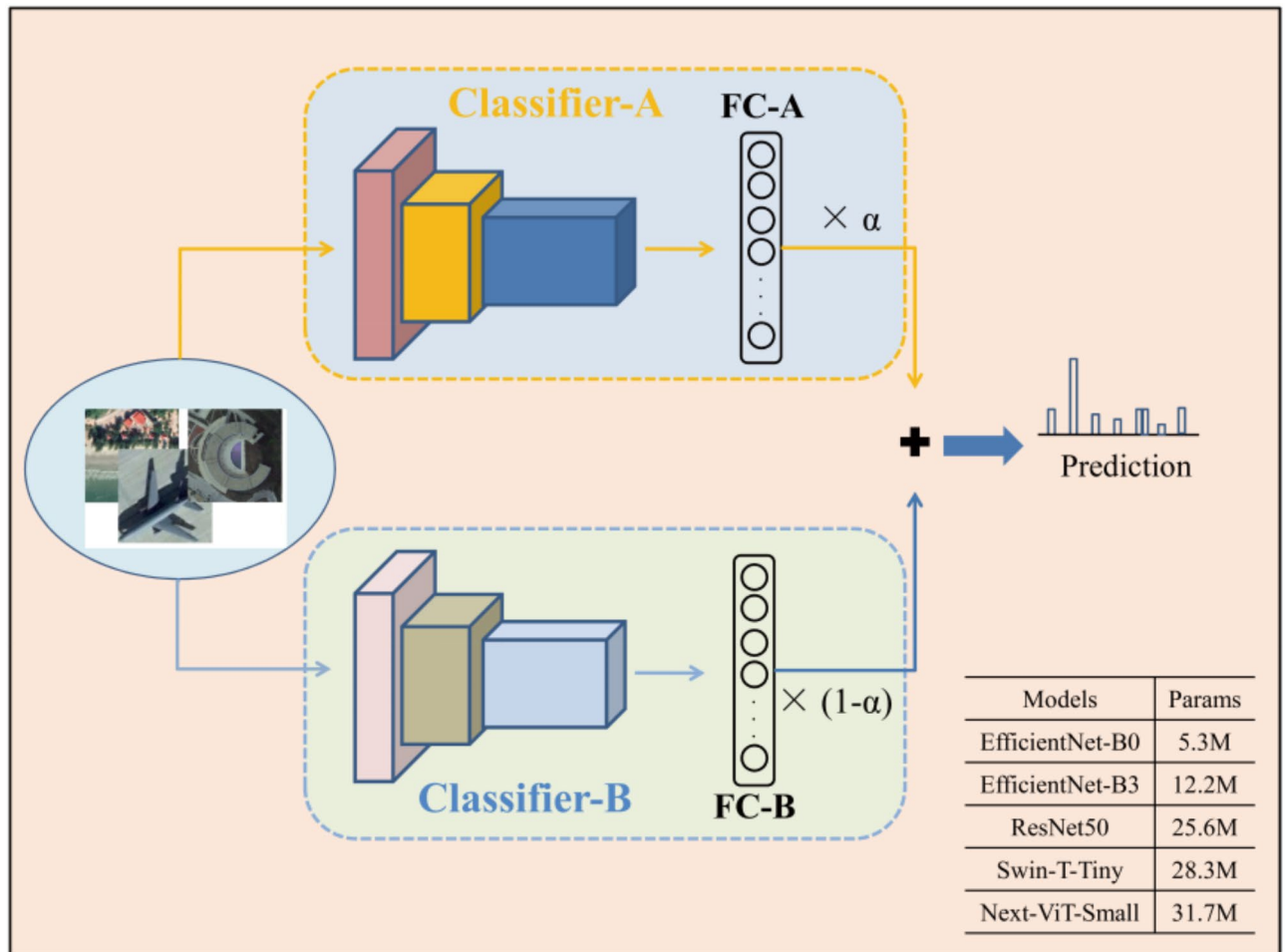


Fig. 4. Architecture of the ESE (Source: Authors own work by Python version 3.9).

Model's architecture

To assess our method, we utilized three CNNs: EfficientNet-B0, EfficientNet-B3, and ResNet-50, in addition to two ViTs: Swin-T-Tiny and N-ViT-Small (N-ViT-S). Detailed descriptions of these five models are available in the Refs. ^{38,39} and ⁴⁰.

In contrast to ResNet-50, both EfficientNet-B0 and -B3 incorporate an internal channel attention structure. ViTs, on the other hand, exhibit self-attention structures combined with position embedding parameters. Furthermore, N-ViT retains convolutional layers as its feature meta-extractors, while Swin-T employs fixed windows for local feature splitting. Consequently, N-ViT demonstrates superior capability in inducing feature biases compared to Swin-T.

In terms of classification performance on ImageNet-1 K, N-ViT-S achieves the highest ranking, followed by Swin-T-Tiny, EfficientNet-B3, -B0, and finally ResNet-50.

Algorithms

Definitions: Let $X_{train} = \{(x_i, y_i)\}$ and X_{test} denote training and testing subsets, respectively. Let f , f_{QAM} , and f_{m-OLS} denote the model in training, the QAM, and the m-OLS function, respectively. Let P denote the model's prediction scores.

Input: Images and labels from training or testing subsets.

Output: Model's accuracy (Acc) results.

```

1:  FOR epoch = 1 TO 60 DO
2:      FOR iteration = 1 TO  $\left(\frac{\text{length}(X_{train})}{32} + 1\right)$  DO
3:          Sample a batch of samples from  $X_{train}$ , and input them to  $f_{QAM}$ , and  $f_{m-OLS}$ .
4:          Obtain  $P$  using the equation:  $P = f(f_{QAM}(x_i))$ .
5:          Compute the loss using  $(P, f_{m-OLS}(y_i))$ .
6:          Update the model's weights through back propagation
7:      END FOR
8:      Compute the model's OA on  $X_{test}$  and save  $Acc$ .
9:  END FOR
10: FOR epoch = 61 TO 300 DO
11:     FOR iteration = 1 TO  $\frac{\text{length}(X_{train})}{32}$  DO
12:         Sample a batch of samples from  $X_{train}$ , and input them to  $f_{QRM}$ , and  $f_{m-OLS}$ .
13:         Obtain  $P$  using the equation:  $P = f(f_{QAM}(x_i))$ .
14:         Compute the loss using  $(P, f_{m-OLS}(y_i))$ .
15:         Update the model's weights through back propagation
16:     END FOR
17:     Compute the model's OA on  $X_{test}$  and save  $Acc$ .
18: END FOR

```

Algorithm 1. Training procedure of CNN or ViT models (pseudocode).

The training algorithm for all CNNs and N-ViT-S is shown in Algorithm 1. The training process spans 300 epochs, with a fixed batch size of 32. However, we introduce a bifurcation in the training process specifically for EfficientNet-B0 to obtain diverse classifiers. Initially, we designate the training epochs from 1 to 60 as Step 1, while the subsequent epochs constitute Step 2. In Step 1, we selectively disable CutMix augmentation or not to yield two distinct groups of EfficientNet-B0 classifiers. In contrast, CutMix remains consistently active during Step 2. The initial learning rate for all CNNs is set to $1E-04$, while for all ViTs, it is $5E-05$. We employ cosine decay as the learning rate scheduler, utilize the Adam-W optimizer, and apply the cross-entropy loss function.

Our experiments indicate that CNNs and N-ViT-S exhibit robust performance due to their ability to induce feature biases. As a result, we disabled the grayscale and auto-contrast functions when training these models. However, we observed that Swin-T converges more slowly than CNNs due to its lesser ability to induce bias. Consequently, we enabled all eight GROs in the QAM when training Swin-T-Tiny, along with modifications to the batch size (increased to 64) and the total training epochs (extended to 600). Additionally, we also implement this enhanced strategy for N-ViT-S on the low-resolution AFGR50 dataset.

Definitions: Let $X_{train} = \{(x_i, y_i)\}$ and $S = \{(s_{1,1}, \dots, s_{i,i})\}$ represent the training subset and the similarity matrix of categories of m-OLS, respectively. Let f_{CM} and N denote the original CutMix function and the hyperparameter for ranking selection, respectively. Let B and N represent a batch of samples from training subsets and the number parameter for ranking selection.

Input: A batch of samples from training subsets.

Output: A batch of cut-and-mixed samples by the ranking selection.

```

1:  FOR each  $x_i$  in  $B$ ,  $i = 1$  to  $\text{length}(B)$  DO
2:      Initialize a similarity list  $L = []$ , which represents the similarity between sample  $x_i$  and the others in  $B$ .
3:      FOR each sample  $x_j$ ,  $j = i+1$  TO  $\text{length}(B)$  DO
4:          Append the similarity score  $s_{ij}$  to the list  $L$ .
5:      END FOR
6:      Obtain the indices of the sorted list  $L$ :  $Index = \text{sorted}(L)$ 
7:      Obtain a sorted sample list:  $L_{Sorted} = B[Index]$ .
8:      Obtain the cut-and-mixed  $x_i$  using the function  $f_{CM}: f_{CM}(x_i, L_{Sorted}(N))$ 
9:  END FOR

```

Algorithm 2. Ranking selection for m-CutMix samples (pseudocode).

The traditional label-smoothing technique typically employs fixed values as soft labels for dataset categories. As a self-adaptive technique, Zhang et al.¹⁰ proposed OLS to reflect the differences in similarity among categories. Initially, the OLS algorithm sets the soft labels to zero and then dynamically updates these labels based on the correctly classified results on training sets. Consequently, it introduces a hyperparameter, denoted as δ , to balance the ground truth (hard) and soft labels before outputting the final labels. Let $Label_H$ and $Label_S$ denote the hard and soft labels, respectively. The final output labels of OLS, denoted as $Label_{OLS}$, can be described as follows:

$$Label_{OLS} = \delta \times Label_S + (1 - \delta) \times Label_H. \quad (3)$$

We employ an empirical value of 0.9 for δ instead of the 0.5 used on ImageNet-1 K, as the similarity among RSIs is higher than that of natural images.

The CutMix algorithm randomly cuts a subarea in a class-A image and then overlaps this patch onto another class-B image. Therefore, if CutMix is used, the cut-and-mixed samples of RSI will contain two category scenes within one image. To guide the model in learning a processed sample, the original CutMix algorithm transforms the sample's label. Let $Label_A$ denote a sample's label from class A, with $Label_B$ representing a sample's label from class B. Let β be the ratio of the cut patch versus its original image. Then, a cut-and-mixed sample's label, $Label_{CM}$, can be described as follows:

$$Label_{CM} = \beta \times Label_A + (1 - \beta) \times Label_B. \quad (4)$$

Moreover, we modified the CutMix algorithm to obtain diverse classifiers for EfficientNet-B3 by using a ranking selection for the cut-and-mixed training samples. As shown in lines 1 to 5 of Algorithm 2, we initially obtain the similarity among a batch of training samples using the similarity matrix from m-OLS. Subsequently, we sort those samples in the batch according to their similarity and employ a ranking selection number, denoted as N , to choose the sample list for cutting patches. Specifically, we set N in m-CutMix equal to 15, or the category number of datasets, to obtain two groups of EfficientNet-B3 classifiers. In other words, we continue to use the original CutMix for training the other models.

Definition: Let X_{test} represent the testing subset. Let C_A and C_B represent the two individual classifiers in the ESE. Let f_{esm} denote the ESE model. Let α be the hyperparameter defined in Equation (2).

Input: Images and labels from testing subsets.

Output: The accuracy (Acc) results of ESE.

```

1: Initialize  $\alpha$  to 0.01.
2: Generate the current ensemble:  $f_{esm} = \alpha \times C_A + (1 - \alpha) \times C_B$ .
3: FOR  $i = 1$  TO 99 DO
4:   Calculate the accuracy of the current ensemble on  $X_{test}$ .
5:   Increment  $\alpha$  by 0.01.
6: END FOR
7: Return the  $Acc$  results along with the corresponding  $\alpha$ .

```

Algorithm 3. Testing procedure of the ESE (pseudocode).

Algorithm 3 provides the process to obtain the final ESE classifier. Specially, we verify the performance of the 99 candidate ensembles with an increase of 0.01 to α per epoch. It is much simpler than other stacking strategies that use extreme gradient boosting to generate the final prediction.

Experiments and results

Dataset introduction

This study utilizes three RSI datasets, specifically AID30, NWPU45, and AFGR50, as benchmarks for assessing the performance of the proposed method. Representative samples from each category within these sets are illustrated in Figs. A1–A3 (located in the Appendix section). As indicated in Table 2, all datasets, with the exception of AID30, exhibit balance across categories. In addition, AFGR50 comprises 50 categories of fine-grained aircraft images, while the remaining datasets consist of classic RSIs depicting ground scenes. Consistent with previous studies, this study also randomly divides all the RSI sets into training and testing subsets.

Dataset	Total categories	Total images	Resolution	Samples per category	TRs
AID30 ⁴¹	30	10,000	600 ²	220–420	20%, 50%
NWPU45 ⁴¹	45	31,500	256 ²	700	10%, 20%
AFGR50 ⁴¹	50	12,500	128 ²	250	10%, 20%, 30%

Table 2. Summary of datasets and TRs.

Evaluation metrics

For performance evaluation, we used two metrics: OA and the confusion matrix. Let N_c and N_t represent the total number of correctly classified samples and the sum of tested samples, respectively. Then, OA is calculated as follows:

$$OA = \frac{N_c}{N_t}. \quad (5)$$

The confusion matrix is a table that contains all category names of a set and provides detailed results per category, including correctly classified and misclassified amounts.

Implementation details

Hyperparameter settings

The models were trained using the Adam-W optimizer with a weight decay of 10^{-6} , and all datasets were processed with an input resolution of 256×256 . The initial learning rate was gradually adjusted throughout the training using a cosine decay schedule. To ensure robustness and reliability, the outcomes of all experiments are presented as averages, calculated from a minimum of three trials.

Experimental environment

The experiments were conducted on three distinct RTX 4070Ti graphics cards, utilizing PyTorch version 2.10.0 in the Ubuntu 20.04 environment. The memory requirement for the Ubuntu 20.04 environment does not exceed 16 GB.

OA results of individual classifiers

In this study, we construct the ESE by utilizing four variations of Algorithm 1 to train EfficientNet-B0 and B3. All modifications to Algorithm 1 originate from setting changes for m-CutMix. For EfficientNet-B0, we derive two classifiers, termed Dual-CutMix-B0 (DC-B0) or Single-CutMix-B0 (SC-B0), by activating the original CutMix in Steps 1 and 2, or only in Step 2, respectively. For EfficientNet-B3, m-CutMix employs two variants: random selection and sorted selection (i.e., N in Algorithm 2 equals 15), with the cut-and-mix operation consistently active. This results in two additional classifiers, termed Random-CutMix-B3 (RC-B3) and Sorted-CutMix-B3 (SC-B3). Additionally, we procured the N-ViT-S and Swin-T-Tiny classifiers using Algorithm 1, with varying training batch sizes and training epochs as introduced in Sect. 3.5.

The OA results of the six models are displayed in Table 3, with the value in bold indicating superior accuracy within a certain column. The results reveal that the two ViTs outperform the other four CNNs on the AID30 dataset, but Efficient-B3 still presents competitive accuracy on the other datasets in comparison with its two ViT competitors. Furthermore, Efficient-B3 even surpasses Swin-T-Tiny on the AFGR50 dataset. AFGR50 consists of low-quality samples at a resolution of 128^2 , while the sample resolution of AID30 is 600^2 . By comparison, the target in the AFGR50 samples is only one aircraft, but samples in AID30 always have multiple targets in a scene. Therefore, Efficient-B3 excels on AFGR50 due to its proficiency at inducing feature biases. On the other hand, ViTs excel on AID30 due to their advantage in capturing global feature dependencies. Consequently, N-ViT-S presents superior and robust accuracy across different datasets due to its CNN-ViT hybrid structure. Similarly, Swin-T-Tiny shows lower OAs on the NWPU45 or AFGR50 due to its limited capability of inducing bias.

Additionally, the OA results from different models on the AID30 (TR-50%) and AFGR50 (TR-30%) are similar. This suggests that models reach saturation when using these larger TRs as indicators. Therefore, a performance comparison using OAs on a smaller TR or on the challenging NWPU45 dataset is more objective.

OA results of ESE

We assessed the ESE method by applying it to enhance five classifiers listed in Table 3. Initially, we introduced three ESEs to augment the performance of RC-B3. These include: RD (RC-B3 with DC-B0), RS (RC-B3 with SC-B0), and DualB3 (RC-B3 with SC-B3). Subsequently, we introduced two ESEs to augment SC-B3, namely SD (SC-B3 with DC-B0) and SS (SC-B3 with SC-B0). In the third step, we introduced DualB0, an ESE to enhance DC-B0, which consists of DC-B0 with DC-B0. In the fourth step, we introduced two ESEs to augment Swin-T-Tiny, namely DNV (DC-B0 with N-ViT-S) and RNV (RC-B3 with N-ViT-S). Finally, we introduced two more ESEs to enhance Swin-T-Tiny, namely DST (DC-B0 with Swin-T-Tiny) and RST (RC-B3 with Swin-T-Tiny).

Classifier	AID30		NWPU45		AFGR50		
	TR-20%	TR-50%	TR-10%	TR-20%	TR-10%	TR-20%	TR-30%
DC-B0	96.90 ± 0.12	98.15 ± 0.04	94.40 ± 0.03	96.03 ± 0.02	90.17 ± 0.15	95.58 ± 0.17	97.12 ± 0.16
SC-B0	96.96 ± 0.14	98.28 ± 0.08	94.43 ± 0.03	96.03 ± 0.06	90.49 ± 0.37	95.66 ± 0.11	97.04 ± 0.11
RC-B3	97.22 ± 0.03	98.27 ± 0.04	94.69 ± 0.03	96.28 ± 0.05	91.91 ± 0.34	96.24 ± 0.12	97.39 ± 0.10
SC-B3	97.31 ± 0.13	98.23 ± 0.04	94.66 ± 0.12	96.24 ± 0.04	90.87 ± 0.05	95.67 ± 0.18	97.14 ± 0.22
Swin-T-Tiny	97.42 ± 0.10	98.39 ± 0.11	94.41 ± 0.19	96.22 ± 0.13	91.72 ± 0.34	96.00 ± 0.24	97.39 ± 0.05
N-ViT-S	97.62 ± 0.11	98.47 ± 0.02	94.81 ± 0.05	96.22 ± 0.07	92.60 ± 0.17	96.33 ± 0.20	97.43 ± 0.07

Table 3. OAs (%) of different classifiers on the three RSI datasets.

Classifier	AID30		NWPU45		AFGR50		
	TR-20%	TR-50%	TR-10%	TR-20%	TR-10%	TR-20%	TR-30%
RC-B3	97.22 ± 0.03	98.27 ± 0.04	94.69 ± 0.03	96.28 ± 0.05	91.91 ± 0.34	96.24 ± 0.12	97.39 ± 0.10
RD-ESE	97.53 ± 0.02	98.45 ± 0.02	95.17 ± 0.05	96.57 ± 0.06	92.21 ± 0.35	96.40 ± 0.18	97.55 ± 0.12
RS-ESE	97.53 ± 0.05	98.44 ± 0.07	95.22 ± 0.06	96.61 ± 0.04	92.32 ± 0.34	96.43 ± 0.16	97.55 ± 0.10
DualB3-ESE	97.47 ± 0.10	98.41 ± 0.06	94.98 ± 0.11	96.44 ± 0.06	91.91 ± 0.32	96.22 ± 0.11	97.41 ± 0.09
SC-B3	97.31 ± 0.13	98.23 ± 0.04	94.66 ± 0.12	96.24 ± 0.04	90.87 ± 0.05	95.67 ± 0.18	97.14 ± 0.22
SD-ESE	97.60 ± 0.14	98.39 ± 0.02	95.13 ± 0.07	96.57 ± 0.03	90.25 ± 0.10	95.58 ± 0.16	97.15 ± 0.19
SS-ESE	97.47 ± 0.10	98.41 ± 0.06	95.16 ± 0.09	96.60 ± 0.04	90.55 ± 0.30	95.65 ± 0.13	97.03 ± 0.13
DC-B0	96.90 ± 0.12	98.15 ± 0.04	94.40 ± 0.03	96.03 ± 0.02	90.17 ± 0.15	95.58 ± 0.17	97.12 ± 0.16
DualB0-ESE	97.13 ± 0.04	98.45 ± 0.04	94.75 ± 0.06	96.28 ± 0.03	90.84 ± 0.29	95.93 ± 0.15	97.24 ± 0.17
Swin-T-Tiny	97.42 ± 0.10	98.39 ± 0.11	94.41 ± 0.19	96.22 ± 0.13	91.72 ± 0.34	96.00 ± 0.24	97.39 ± 0.05
DST-ESE	97.75 ± 0.11	98.63 ± 0.06	95.24 ± 0.03	96.65 ± 0.05	92.19 ± 0.31	96.40 ± 0.21	97.64 ± 0.09
RST-ESE	97.86 ± 0.12	98.59 ± 0.05	95.29 ± 0.04	96.78 ± 0.02	92.97 ± 0.37	96.60 ± 0.21	97.67 ± 0.04
N-ViT-S	97.62 ± 0.11	98.47 ± 0.02	94.81 ± 0.05	96.22 ± 0.07	92.60 ± 0.17	96.33 ± 0.20	97.43 ± 0.07
DNV-ESE	97.75 ± 0.11	98.59 ± 0.04	95.27 ± 0.01	96.59 ± 0.07	92.86 ± 0.10	96.59 ± 0.15	97.75 ± 0.07
RNV-ESE	97.86 ± 0.12	98.59 ± 0.15	95.34 ± 0.08	96.70 ± 0.04	93.27 ± 0.29	96.77 ± 0.24	97.80 ± 0.01

Table 4. OAs (%) of different ESE classifiers on the three RSI datasets.

Classifier	AID30		NWPU45		AFGR50	
	TR-20%	TR-10%	TR-20%	TR-10%	TR-20%	TR-10%
RD-ESE	97.53 ± 0.02	95.17 ± 0.05	96.57 ± 0.06	92.21 ± 0.35	96.40 ± 0.18	
RD-ESE-9 C	97.50 ± 0.03 ↓ 0.03	95.16 ± 0.05 ↓ 0.01	96.57 ± 0.07 ↓ 0.0	92.17 ± 0.39 ↓ 0.04	96.38 ± 0.19 ↓ 0.02	
DualB0-ESE	97.13 ± 0.04	94.75 ± 0.06	96.28 ± 0.03	90.84 ± 0.29	95.93 ± 0.15	
DualB0-ESE-9 C	97.11 ± 0.03 ↓ 0.02	94.74 ± 0.06 ↓ 0.01	96.26 ± 0.05 ↓ 0.02	90.81 ± 0.30 ↓ 0.03	95.91 ± 0.19 ↓ 0.02	
RST-ESE	97.86 ± 0.12	95.29 ± 0.04	96.78 ± 0.02	92.97 ± 0.37	96.60 ± 0.21	
RST-ESE-9 C	97.81 ± 0.10 ↓ 0.05	95.26 ± 0.06 ↓ 0.03	96.73 ± 0.03 ↓ 0.05	92.93 ± 0.42 ↓ 0.04	96.59 ± 0.27 ↓ 0.01	
RNV-ESE	97.86 ± 0.12	95.34 ± 0.08	96.70 ± 0.04	93.27 ± 0.29	96.77 ± 0.24	
RNV-ESE-9 C	97.85 ± 0.14 ↓ 0.01	95.31 ± 0.11 ↓ 0.03	96.68 ± 0.05 ↓ 0.02	93.24 ± 0.33 ↓ 0.03	96.76 ± 0.25 ↓ 0.01	

Table 5. OAs (%) of different ESE-9 C classifiers on the three RSI datasets.

As depicted in Table 4, we organized the OA values of the proposed ESE classifiers into five sections, separated by bold black lines. The bold values signify the highest OA within a column of these sections. In the first section, it is evident that RC-B3 can achieve substantial OA enhancements by collaborating with other EfficientNet-B3 or -B0 models. However, ESEs comprising different architectures, such as RD-ESE or RS-ESE, exhibit superior OA values. This observation establishes the first principle of our ESE method: incorporating two classifiers with varied architectures in ESE results in better performance. In the second section, it is clear that the ESE method can also augment the accuracy of SC-B3. Yet, SC-B3 displays lower OAs on the AFGR50 compared to RC-B3. As a result, SD-ESE and SS-ESE also demonstrate lower OAs on the AFGR50. This observation establishes the second principle of our ESE method: the inclusion of a more robust classifier in ESE leads to better performance.

In Tables 3, 4 and 5, the OA results validate the aforementioned principles once again. Comparatively, the ESE classifiers comprising N-ViT-S exhibit superior OAs than those comprising Swin-T-Tiny. However, our ESE method amplifies the OAs of Swin-T-Tiny more significantly than those of N-ViT-S, though Swin-T-Tiny has a limited capability for inducing feature biases. Therefore, this observation validates that our ESE method is an effective and efficient strategy to amalgamate the benefits of CNNs and ViTs.

Sensitivity analysis

The process of generating ESEs entails two pivotal decisions that can significantly influence the performance of the method. Initially, we implement an optimal search among 99 potential ESEs, a step that corresponds to substantial time expenditure during training. Subsequently, we restricted our use to only the EfficientNet-B0 and -B3 models as CNN classifiers in previous experiments. This restriction, however, limits the generality of the ESE method when other CNNs are excluded from the method's framework. To address these limitations, we conducted two supplementary experiments with two primary objectives: firstly, to curtail the time expenditure associated with generating ESEs, and secondly, to verify the applicability of generating ESEs using another CNN.

First, we select only nine candidate ESEs (denoted by a suffix: 9 C) by setting α in Eq. (2) to increase from 0.1 to 0.9 at a 0.1 interval. We present the corresponding OA results in Table 5. In the table, we omit the values of the TRs where models reach saturation, while the values in bold denote a decreased OA value. As Table 5 reveals, the accuracy losses of all these ESE classifiers obtained by our nine-candidate strategy are minimal, ranging

Classifier	AID30	NWPU45		AFGR50	
	TR-20%	TR-10%	TR-20%	TR-10%	TR-20%
RD-ESE	97.53 ± 0.02	95.17 ± 0.05	96.57 ± 0.06	92.21 ± 0.35	96.40 ± 0.18
RD-ESE-R50	97.50 ± 0.09↓ 0.03	94.95 ± 0.06↓ 0.22	96.46 ± 0.06↓ 0.11	91.94 ± 0.37↓ 0.27	96.24 ± 0.13↓ 0.16
DualB0-ESE	97.13 ± 0.04	94.75 ± 0.06	96.28 ± 0.03	90.84 ± 0.29	95.93 ± 0.15
DualB0-ESE-R50	97.20 ± 0.06↑ <i>0.07</i>	94.80 ± 0.06↑ <i>0.05</i>	96.29 ± 0.04↑ <i>0.01</i>	90.24 ± 0.16↓ 0.60	95.58 ± 0.18↓ 0.35
RST-ESE	97.86 ± 0.12	95.29 ± 0.04	96.78 ± 0.02	92.97 ± 0.37	96.60 ± 0.21
RST-ESE-R50	97.56 ± 0.15↓ 0.30	94.81 ± 0.07↓ 0.48	96.43 ± 0.07↓ 0.35	91.73 ± 0.37↓ 1.24	96.04 ± 0.28↓ 0.56
RNV-ESE	97.86 ± 0.12	95.34 ± 0.08	96.70 ± 0.04	93.27 ± 0.29	96.77 ± 0.24
RNV-ESE-R50	97.65 ± 0.14↓ 0.21	94.94 ± 0.03↓ 0.40	96.39 ± 0.02↓ 0.31	92.62 ± 0.18↓ 0.65	96.37 ± 0.24↓ 0.40

Table 6. OAs (%) of different ESE-R50 classifiers on the three RSI datasets.

from 0.0 to 0.05%. This observation demonstrates the robustness of the ESE method against variations in α . Consequently, we can adopt the nine-candidate strategy to reduce time costs by 91% during the verification of ESE classifiers, with a negligible sacrifice in accuracy of up to 0.05%.

In addition, we replaced the EfficientNet-B0 or -B3 model with a Resnet50 model in various ESEs such as RD-ESE, DuanlB0-ESE, RST-ESE, or RNV-ESE. Table 6 illustrates the OA comparison for these ESE classifiers, now based on Resnet50 (denoted by a suffix: R50), divided into four unique sections. We excluded the TRs where models reached saturation, and the values highlighted in bold or italics represent a decrease or increase in OA, respectively.

The first half of the sections show that substituting Efficient-B0 with Resnet50 results in a slight OA improvement for DuanlB0-ESE on three datasets, except for the AFGR50, while similar OA enhancements for RD-ESE are negligible. On the contrary, substituting Efficient-B3 with Resnet50 within RST-ESE or RNV-ESE leads to more noticeable reductions in OA, as depicted in the last two sections of Table 6. Most importantly, this substitution using Resnet50 leads all four ESEs to show significant OA decreases on the low-resolution AFGR50 dataset, especially when Swin-T-Tiny, with its limited capability for inducing feature biases, is included in the ESE model.

Despite this, all these Resnet50-based ESEs still exhibit improved OAs when compared to the single-CNN or -ViT classifiers in Table 3. Therefore, this observation indicates that our ESE method retains its effectiveness when different CNNs are incorporated into the ESE model. However, the ESE method can achieve better performance, particularly when it leverages the benefits of superior CNNs to enhance its ViT partner.

Performance comparison with previous methods

This study conducts an exhaustive performance comparison of the proposed models with 48 cutting-edge methods published between 2020 and 2024. The comparative results are encapsulated in Tables 7, 8 and 9. Within these tables, the term ‘None’ signifies the absence of detailed information in the corresponding literature, while ‘>’ denotes values estimated by the base model. Furthermore, values highlighted in bold indicate superior performance within a specific column.

Results on AID30

As Table 7 illustrates, two-thirds of the previous methods achieved OAs below 96.0% when using the TR of 20% on the AID30 dataset as a test bed. In contrast, our ESE, CNN, and ViT models demonstrate significantly higher accuracy, reaching up to 97.9%, 97.2%, and 97.6%, respectively. Only two previous methods that employ dual Swin-T models, specifically TSTNet and IBSwin-CR, have achieved accuracies closely matched with our CNN or ViT models. However, these two methods both have an excessively large model size, which hinders inference speeds. Therefore, our ViT or ESE models can perform better with an 80% reduction in parameters compared to their top competitors.

Our methods maintain strong performance at TR-50%, although several previous methods achieve similar OA values. However, deep learning models tend to reach saturation when sufficient training samples are available, making TR-20% a more objective measure of performance. Therefore, using the TR-20% as a performance evaluator is more objective. Nonetheless, the OA values reveal that most previous methods still present OAs below 98.0% at TR-50%, even when employing multiple models. Thus, the comparison on the AID30 dataset also underscores the advantage of our methods, particularly when the availability of training samples is constrained.

Results on NWPU45

As Table 8 illustrates, all of our models achieve an accuracy exceeding 94.4% when using a TR of 10% on the NWPU45 as a benchmark. Specifically, the OA of the ESE models reaches up to 95.3%, surpassing N-ViT-S and Swin-T-Tiny with 0.5% and 0.9% improvements, respectively. In contrast, 34 out of 39 previous methods yield an OA value below 93.5%, while only P2FEViT achieves an OA value of 94.9%. Moreover, TSTNet and IBSwin-CR, which perform well on the AID30, both yield inferior OAs below 94.1%.

Similarly, our models both achieve an accuracy exceeding 96.0% when using a TR of 20% as a performance benchmark. Notably, our RST-ESE model yields OAs up to 96.8%, surpassing the top-ranked P2FEViT or TSTNet with approximately a 1.1% OA improvement. In comparison, most previous methods yield OAs below

Method	Technique feature	Params (M)	Published year	OA (%)		
				TR-20%	TR-50%	
SLGE-CNN ⁷	Single CNN	5.1	TGRS2022	96.10 ± 0.18 (TR-80%)		
AF-CNN ⁸		3.8	TGRS2022	95.96 (TR-60%)		
LCNN-HWCF ¹¹		0.6	RS2022	95.76 ± 0.16	97.43 ± 0.28	
SCCNN ⁴²		0.5	RS2022	93.15 ± 0.25	97.31 ± 0.10	
MF2CNet ⁴³		33.2	TGRS2022	95.54 ± 0.17	97.02 ± 0.28	
ViT-Base ⁴⁴	Single ViT	86.6	RS2021	95.86 ± 0.28	None	
ET-GSNet ¹²		98.3	TGRS2022	95.58 ± 0.18	96.88 ± 0.19	
ViTAEv2 ¹³	Single Swin-T	27.6	TGRS2023	96.81 ± 0.03	98.30 ± 0.04	
EfficientNet-B3-Attn ⁴⁵	Attention module for CNN	> 12.0	ACCESS2021	94.45 ± 0.76	96.56 ± 0.12	
MBLANet ⁴⁶		> 25.6	TIP2022	95.60 ± 0.17	97.14 ± 0.03	
EAM-CNN ¹⁴		> 46.8	GRSL2021	94.26 ± 0.11	97.06 ± 0.19	
LHNet ⁴⁷	Feature fusion	> 46.8	TGRS2022	93.30 ± 0.10	97.81 ± 0.13	
SCViT ¹⁵		40.1	TGRS2022	95.56 ± 0.17	96.98 ± 0.16	
MLF2Net ⁴⁸		23.8	GRSL2022	95.44 ± 0.25	97.08 ± 0.17	
SEMSDNet ⁴⁹		3.7	JSTARS2021	94.23 ± 0.63	97.64 ± 0.51	
LmNet ⁵⁰		> 25.0	ACCESS2021	95.82 ± 0.25	97.12 ± 0.14	
D-CNN ⁵¹		None	RS2021	94.63	96.43	
GCSANet ¹⁶	Multiple models	8.1	JSTARS2022	95.96 ± 0.38	97.53 ± 0.32	
ACGLNet ¹⁷		33.6	RS2022	94.44 ± 0.09	96.10 ± 0.10	
SF-MSFormer ¹⁸		36.3	TGRS2023	None	98.72 ± 0.31	
AGOS ⁵²		None	TGRS2022	95.81 ± 0.25	97.43 ± 0.21	
GRMA-Net ¹⁹		54.1	TGRS2022	96.19 ± 0.48	97.84 ± 0.39	
ACNet ⁵³		> 276.6	JSTARS2021	93.33 ± 0.29	95.38 ± 0.29	
T-CNN ⁵⁴		15.9	TGRS2022	94.55 ± 0.27	96.72 ± 0.23	
GLDBS ⁵⁵		> 23.4	GRSL2022	95.45 ± 0.19	97.01 ± 0.22	
TRS ⁵⁶		46.3	RS2021	95.54 ± 0.18	98.48 ± 0.06	
CTNet ⁵⁷		> 107.8	GRSL2022	96.25 ± 0.10	97.70 ± 0.11	
HHTL ⁵⁸		> 173.2	JSTARS2022	95.62 ± 0.13	96.88 ± 0.21	
L2RCF ⁵⁹		46.7	TGRS2023	97.00 ± 0.17	97.80 ± 0.22	
ViT-CL ⁶⁰		Custom learning framework	86.0	JSTARS2023	95.60	97.42
GSCCTL ⁶¹			None	IJRS2022	91.32	None
MGDNet ⁶²	None		TGRS2023	86.52 ± 0.81	None	
TSTNet ²¹	Multiple Swin-Ts	173.0	RS2002	97.20 ± 0.22	98.70 ± 0.12	
IBSwin-CR ²²		164.0	JSTARS2023	97.61 ± 0.12	98.78 ± 0.09	
MFST ⁶³		None	GRSL2022	96.23 ± 0.16	97.38 ± 0.08	
mmsCNN-HMM ²⁷	Multi-CNN ensemble	19.0	RS2022	93.93 ± 0.15	97.81 ± 0.04	
MGML ²⁸		None	TNNLS2023	94.47 ± 0.15	97.89 ± 0.07	
ESD-MBENet ²⁹		23.9	TGRS2022	96.39 ± 0.21	98.40 ± 0.23	
RC-B3	Single CNN	12.2	This work	97.22 ± 0.03	98.27 ± 0.04	
Swin-T-Tiny	Single ViT	28.3		97.42 ± 0.10	98.39 ± 0.11	
N-ViT-S		31.7		97.62 ± 0.11	98.47 ± 0.02	
RD-ESE	Dual-CNN ensemble	17.5		97.53 ± 0.02	98.45 ± 0.02	
RST-ESE	CNN-ViT hybrid ensemble	40.5		97.86 ± 0.12	98.59 ± 0.05	
RNV-ESE		43.9		97.86 ± 0.12	98.59 ± 0.15	

Table 7. OA results across various methods on the AID30 dataset.

95%. This phenomenon substantiates that our models consistently exhibit robust generalization capabilities when utilizing different TRs on either the AID30 or NWPU45 datasets as benchmarks.

P2FEViT employs a CNN-ViT hybrid architecture with a substantial model size of approximately 112 million parameters. Consequently, P2FEViT presents a slight OA improvement over our ViT models at a TR of 10% due to its hybrid structure. However, our hybrid ESE models not only exhibit significant OA improvements of up to 1.1% but also demonstrate a 70% reduction in parameters when compared to P2FEViT. Therefore, this comparison underscores that our ESE method is a more efficient strategy to leverage the strengths of both CNNs and ViTs.

Method	Technique feature	Params (M)	Published year	OA (%)		
				TR-10%	TR-20%	
TPENAS-CNN ⁶	Single CNN	1.8	RS2023	None	90.38	
SLGE-CNN ⁷		5.1	TGRS2022	96.44 ± 0.21 (TR-80%)		
AF-CNN ⁸		3.8	TGRS2022	95.32 (TR-60%)		
LCNN-HWCF ¹¹		0.6	RS2022	93.10 ± 0.12	94.53 ± 0.25	
SCCNN ⁴²		0.5	RS2022	92.02 ± 0.50	94.39 ± 0.16	
MF2CNet ⁴³		33.2	TGRS2022	92.07 ± 0.22	93.85 ± 0.27	
ViT-Base ⁴⁴	Single ViT	86.6	RS2021	93.83 ± 0.46	None	
ET-GSNet ¹²		98.3	TGRS2022	92.72 ± 0.28	94.50 ± 0.18	
ViTAEv2 ¹³	Single Swin-T	27.6	TGRS2023	94.41 ± 0.11	95.60 ± 0.06	
MBLANet ⁴⁶	Attention module for CNN	>25.6	TIP2022	92.32 ± 0.15	94.66 ± 0.11	
EAM-CNN ¹⁴		>46.8	GRSL2021	91.91 ± 0.22	94.29 ± 0.09	
LHNet ⁴⁷	Feature fusion	>46.8	TGRS2022	89.89 ± 0.15	92.53 ± 0.13	
SCViT ¹⁵		40.1	TGRS2022	92.72 ± 0.04	94.66 ± 0.10	
MLF2Net ⁴⁸		23.8	GRSL2022	92.35 ± 0.17	94.84 ± 0.09	
SEMSDNet ⁴⁹		3.7	JSTARS2021	91.68 ± 0.39	93.89 ± 0.63	
LmNet ⁵⁰		>25.0	ACCESS2021	93.00 ± 0.11	94.85 ± 0.14	
MLFC-Net ⁶⁴		65.2	CG2022	92.52 ± 0.38	94.76 ± 0.08	
D-CNN ⁵¹	Multiple models	None	RS2021	89.88	94.44	
GCSANet ¹⁶		8.1	JSTARS2022	93.39 ± 0.39	94.95 ± 0.36	
SF-MSFormer ¹⁸		36.3	TGRS2023	92.74 ± 0.23	94.83 ± 0.13	
AGOS ⁵²		>12.5	TGRS2022	93.04 ± 0.35	94.91 ± 0.17	
MGSN ⁶⁵		>12.0	JSTARS2022	91.92 ± 0.12	94.33 ± 0.08	
GRMA-Net ¹⁹		54.1	TGRS2022	93.67 ± 0.21	95.32 ± 0.28	
ACNet ⁵³		>276.6	JSTARS2021	91.09 ± 0.13	92.42 ± 0.16	
T-CNN ⁵⁴		15.9	TGRS2022	90.25 ± 0.14	93.05 ± 0.12	
GLDBS ⁵⁵		>23.4	GRSL2022	92.24 ± 0.21	94.46 ± 0.15	
TRS ⁵⁶		46.3	RS2021	93.06 ± 0.11	95.56 ± 0.20	
CTNet ⁵⁷		>107.8	GRSL2022	93.90 ± 0.14	95.40 ± 0.15	
HHTL ⁵⁸		>173.2	JSTARS2022	92.07 ± 0.44	94.21 ± 0.09	
P2FEViT ²⁰		>112.2	RS2023	94.97 ± 0.13	95.74 ± 0.19	
L2RCF ²⁰		Custom learning framework	46.7	TGRS2023	94.58 ± 0.16	95.60 ± 0.12
ViT-CL ⁶¹			86.0	JSTARS2023	92.85	94.69
GSCCTL ⁶²			None	IJRS2022	91.96	None
MGDNet ⁶⁶	None		TGRS2023	84.81 ± 0.36	91.41 ± 0.69	
LGRIN ⁶⁷	4.6		TGRS2022	91.91 ± 0.15	94.43 ± 0.16	
TSTNet ²¹	Multiple Swin-Ts		173.0	RS2002	94.08 ± 0.24	95.70 ± 0.10
IBSwin-CR ²²		164.0	JSTARS2023	93.98 ± 0.24	95.65 ± 0.11	
MFST ⁶³		None	GRSL2022	92.64 ± 0.08	94.90 ± 0.06	
Hydra ⁶⁸	Multi-CNN ensemble	331.0	TGRS2019	92.44 ± 0.34	94.51 ± 0.21	
mmsCNN-HMM ³⁸		19.0	RS2022	93.43 ± 0.25	95.51 ± 0.21	
MGML ³⁹		None	TNNLS2023	90.69 ± 0.14	93.36 ± 0.12	
ESD-MBENet ²¹		23.9	TGRS2022	93.05 ± 0.18	95.36 ± 0.14	
RC-B3	Single CNN	12.2	This Work	94.69 ± 0.03	96.28 ± 0.05	
Swin-T-Tiny	Single ViT	28.3		94.41 ± 0.19	96.22 ± 0.13	
N-ViT-S		31.7		94.81 ± 0.05	96.22 ± 0.07	
RD-ESE	Dual-CNN ensemble	17.5		95.17 ± 0.05	96.57 ± 0.06	
RST-ESE	CNN-ViT hybrid ensemble	40.5		95.29 ± 0.04	96.78 ± 0.02	
RNV-ESE		43.9		95.34 ± 0.08	96.70 ± 0.04	

Table 8. OA results across various methods on the NWP45 dataset.

Results on AFGR50

Table 9 presents the comparison of OA on the AFGR50 dataset, demonstrating that MBC-Net holds the highest OA among previous methods. However, MBC-Net exhibits gaps in OA of 1.6–2.3% when compared to our ViT and ESE methods at a TR of 10%. As the TR increases to 20%, MBC-Net narrows the OA gaps to between 0.8%

Method	Technique feature	Params (M)	Published year	OA (%)		
				TR-10%	TR-20%	TR-30%
ResNet50 ⁴	Single CNN	25.6	IASC2023	89.85 ± 0.63	95.11 ± 0.24	96.58 ± 0.13
LS-EfficientNet ⁵		5.3	AETIC2023	90.39 ± 0.51	95.63 ± 0.21	97.03 ± 0.19
P2FEViT ²⁰	Multiple models	> 91.9	RS2023	89.24 ± 0.10	95.22 ± 0.13	97.27 ± 0.15
MBC-Net ⁶⁹		17.3	IJICC2024	91.01 ± 0.61	96.13 ± 0.26	97.28 ± 0.27
RC-B3	Single CNN	12.2	This work	91.91 ± 0.34	96.24 ± 0.12	97.39 ± 0.10
Swin-T-Tiny	Single ViT	28.3		91.72 ± 0.34	96.00 ± 0.24	97.39 ± 0.05
N-ViT-S		31.7		92.60 ± 0.17	96.33 ± 0.20	97.43 ± 0.07
RD-ESE	Dual-CNN ensemble	17.5		92.21 ± 0.35	96.40 ± 0.18	97.55 ± 0.12
RST-ESE	CNN-ViT hybrid ensemble	40.5		92.97 ± 0.37	96.60 ± 0.21	97.67 ± 0.04
RNV-ESE		43.9		93.27 ± 0.29	96.77 ± 0.24	97.80 ± 0.01

Table 9. OA results across various methods on the AFGR50 dataset.

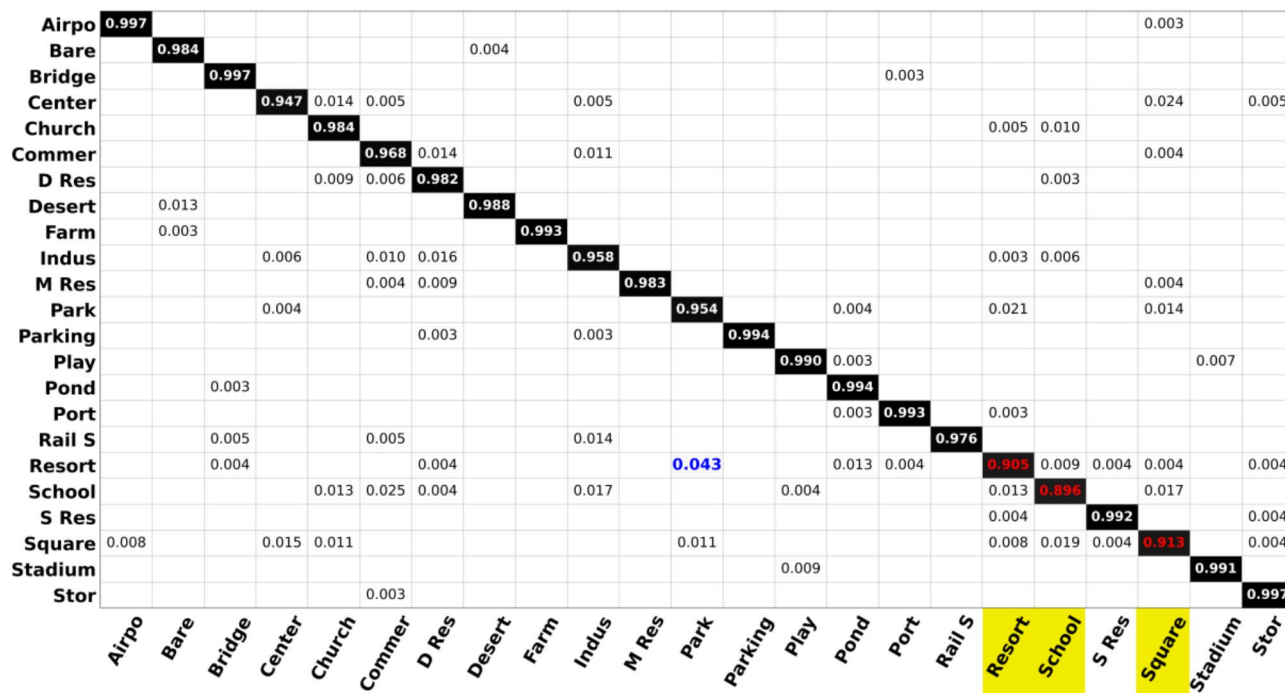


Fig. 5. The confusion matrix of RNV-ESE on the AID30 dataset (TR=20%) (Source: Authors own work by Python version 3.9).

and 1.6%. When models reach saturation at TR-30%, P2FEViT also exhibits OAs comparable to our single-model methods, yet still presents significant OA gaps in comparison to our ESE methods.

In conclusion, based on the OA results depicted in Tables 7, 8 and 9, it is clear that our QA strategy is highly competitive for enhancing CNN or ViT in classifying RSIs. Generally, in comparison to previous state-of-the-art methods in the literature, our single-CNN or single-ViT models provide advantages in terms of accuracy and efficiency. Furthermore, our ESE method takes a step further to offer a lightweight and beginner-friendly approach for integrating the benefits of CNNs and ViTs, which demonstrates further improvements in accuracy while maintaining efficiency and simplicity.

Confusion matrixes

In this section, we introduce the confusion matrix figures (CMFs) of RNV-ESE and RST-ESE to assess performance across various categories. We apply the CMFs at a TR of 20% on the AID30, NWPU45, and AFGR50 datasets. Figures 5, 6 and 7 display the CMFs for RNV-ESE, while we present those of RST-ESE in Figs. A5–A7 in the Appendix. In these CMFs, we interpret a value of 1.0 as an OA of 100%. Category names highlighted in yellow or values in bold indicate the most confusing class (referring to categories with low accuracy scores) in a given dataset. We use blue values to signify a misclassified ratio (the percentage of all class-A samples misclassified as class-B) that exceeds 3.0%. For enhanced readability, we exclude categories that achieve a 100% OA on the AID30 or an OA above 98% on the NUPU45 and AFGR50 from the CMFs.

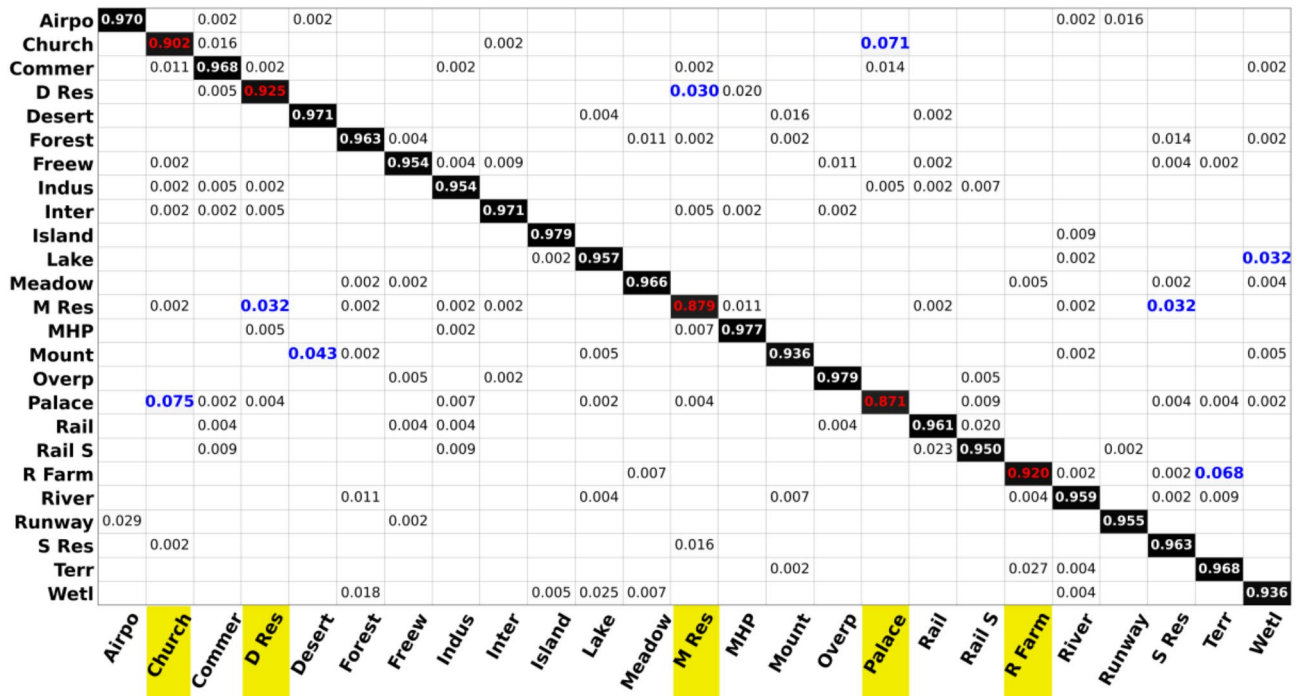


Fig. 6. The confusion matrix of RNV-ESE on the NWPU45 dataset (TR=20%) (Source: Authors own work by Python version 3.9).

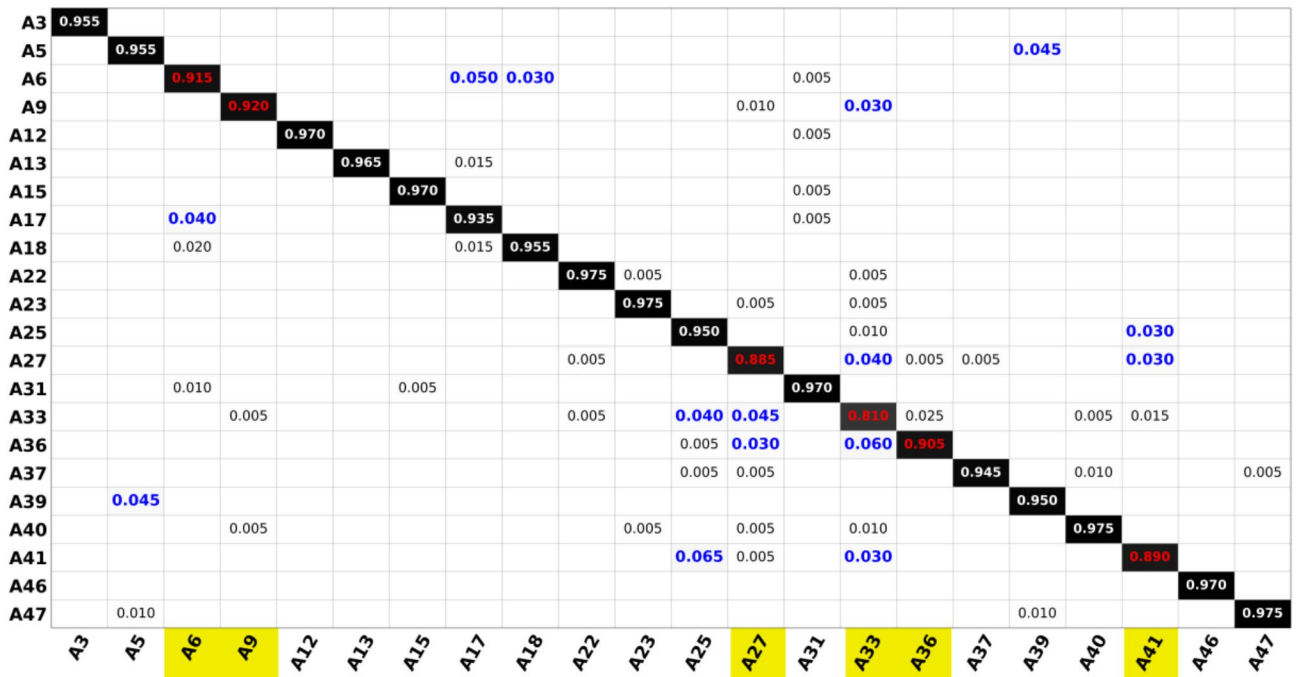


Fig. 7. The confusion matrix of RNV-ESE on the AFGR50 dataset (TR=20%) (Source: Authors own work by Python version 3.9).

As Fig. 5 illustrates, 23 categories within AID30 have an OA below 100%, but only 8 of these 23 classes have an OA below 98.0%. More specifically, only three categories, including the most confusing ones of resort, school, and square, present OAs lower than 93.0%. Additionally, only the park category has a misclassification ratio larger than 3.0%. This observation reveals that less than one-third of categories exhibit high inter-class similarity, and the most confusing ones pose the key challenge within AID30. The sample numbers of the resort and school classes are 290 and 300, respectively, both below the average value of 333 within all categories. Therefore, this

CMF of AID30 further substantiates that using the TR of 20% on AID30 as a test bed is more convincing than the TR of 50%.

As depicted in Fig. 6, 25 categories within NWPU45 have OAs below 98.0%, with 7 and 5 of these 25 classes presenting OAs below 94% and 93.0%, respectively. Additionally, 8 categories in this CMF have a misclassification ratio larger than 3.0%, with three of them having a high misclassification value, reaching up to 7.5%. These observations indicate that the inter-class similarity and intra-class dissimilarity within NWPU45 are more pronounced compared to AID30.

The categories within NWPU45 that pose significant challenges include the church, dense residential, medium residential, palace, and rectangular farmland. Additionally, the mountain and wetland classes register an OA below 94.0%. In comparison, the challenging classes within AID30 do not include nature scenes. This difference indicates that the inherent features of RSIs include not only noise backgrounds associated with human settlements but also a state of low-quality imaging. As a result, the NWPU45 dataset serves as a more credible benchmark for performance evaluation.

Figure 7 shows that 22 categories within AFGR50 present OAs below 98.0%, and six categories have OAs under 93.0%. Additionally, nine categories exhibit a misclassification ratio above 3%, as the blue indicators denote. AFGR50 is a fine-grained dataset that includes various types of aircraft at a resolution of 128^2 . However, each sample in AFGR50 features a plain background. As a result, adverse imaging conditions pose the main challenge for accurate recognition within AFGR50. Therefore, the CMF for AFGR50 highlights the significant impact of imaging conditions on model performance.

Comparing our RNV-ESE with earlier methods, similar categories of confusion are evident across the AID30, NWPU45, and AFGR50 datasets. However, RNV-ESE exhibits enhancements in OA across all categories, not just a specific class. Some prior methods have excelled in certain AID30 categories but fell short in the more challenging NWPU45 categories. For example, the square category of AID30 shows a high OA with the TRS method. This suggests that some previous methods lack consistent performance across various RSI datasets.

From all CMFs and comparisons, it's clear that the main source of RNV-ESE's performance improvements is the superior quality of the training data. Our QA strategy effectively aligns the data distribution features between training and testing subsets, leading to an overall enhancement in model performance across all categories.

Visualization and analysis

We use two visualization methods to highlight two critical aspects of the proposed models: activation mappings and feature effectiveness. We initially use the gradient-weighted class activation mapping (Grad-CAM)⁷¹ to visualize the essential information within the model predictions. We then use the t-Distributed Stochastic Neighbor Embedding, commonly abbreviated as t-SNE, to demonstrate the effectiveness of the model features.

Grad-CAM results

Figure 8 presents the Grad-CAM figures (GCFs) across ten distinct RSI categories, with the first through ninth categories corresponding to three particularly challenging classes from the AFGR50, AID30, and NWPU45 datasets, respectively. The final category, representing the river, is a typical scene dominated by long-range watercourses. These GCFs offer a detailed view of how different models process and highlight key areas within the scenes, providing insights into the models' varying strategies for feature extraction.

The second-to-last row in the figure displays the GCFs for the RC-B3 model. These GCFs reveal that RC-B3 focuses on local features of ground targets, such as the wings of aircraft or the rounded roofs of buildings. These localized features are highlighted with brighter colors, indicating higher activation levels. While RC-B3

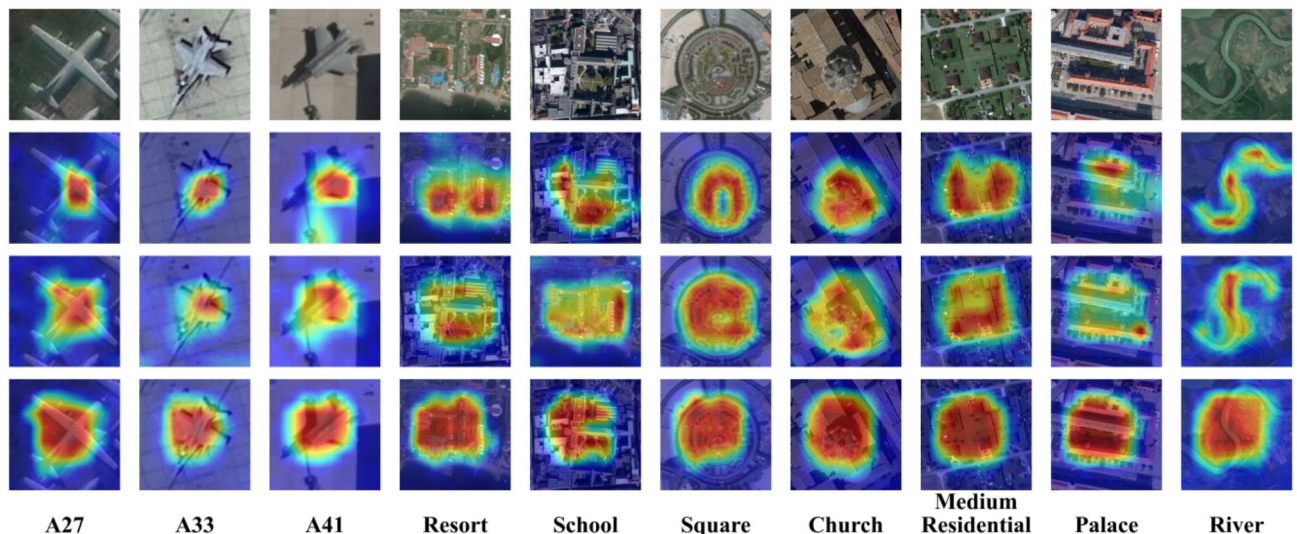


Fig. 8. Grad-CAM visualizations of models on three datasets (Source: Authors own work by Python version 3.9).

is effective at identifying these key visual patterns within specific RSIs, its reliance on convolutional operations limits its ability to capture long-range dependencies, which could be crucial for understanding more complex scenes. This limitation is particularly evident in scenarios where the relationships between distant features are critical for accurate recognition.

In contrast, the third-row GCFs for the N-ViT-S model demonstrate a more sophisticated approach. While N-ViT-S still identifies local features as key activation areas, it goes a step further by capturing long-range dependencies between these features. This is especially noticeable in the resort and river categories, where N-ViT-S exhibits clearer distinctions compared to RC-B3. The model's ability to integrate global context through its vision transformer architecture enables it to recognize more complex relationships within the scene, improving its performance over traditional convolution-based models like RC-B3.

The final row of GCFs showcases the Swin-T-Tiny model, which introduces a unique feature extraction strategy. Instead of relying on convolutional operations like RC-B3 or the transformer mechanisms of N-ViT-S, Swin-T-Tiny utilizes window-based splitting operations. This allows the model to first capture local features and then process dependencies between them. Swin-T-Tiny places significant emphasis on the central regions of ground targets, which can be crucial for accurate scene recognition. Its ability to handle both local and global dependencies through this window-based mechanism enables it to achieve a more comprehensive understanding of complex RSIs, particularly in large or intricate scenes.

Overall, the GCFs vividly illustrate the architectural differences in feature extraction strategies across the models. RC-B3 excels at local feature recognition but struggles with long-range dependencies. N-ViT-S effectively integrates the strengths of both CNNs and ViTs, balancing the need for local feature extraction with the ability to capture global relationships. Swin-T-Tiny, with its window-based approach, focuses on central features and dependencies, offering a more holistic understanding of complex scenes. These differences underscore that N-ViT-S, by leveraging both convolutional and transformer architectures, demonstrates superior performance in recognizing both high- and low-resolution RSIs, positioning it as the most robust model among the three for diverse input data.

t-SNE results

The t-SNE clustering figure (TCF), a bi-dimensional projection of post-principal component analysis processed model features, is illustrated in Fig. 9. It shows that all AID30 categories are distinctly spaced. Yet, two areas (marked by a red rectangle) contain samples that overlap from four classes. Rectangle A includes the commercial and school classes, while rectangle B pertains to the resort with parking. Earlier confusion matrices have suggested a higher similarity between these four categories. Therefore, this observation aligns consistently with the TCF representation.

As illustrated in Fig. 10, all categories within NWPU45 are distinctly separated, but six classes within three red rectangles contain overlapping samples. Rectangle A encompasses the dense and medium residential categories; rectangle B includes the church and palace; and rectangle C comprises the railway and the railway station. This observation also aligns with the previously identified challenging categories presented by confusion matrices.

Figure 11 illustrates that all categories within AFGR50 are distinctly separated. Yet, the TCF uncovers that six categories have overlapping samples, as indicated by the red rectangles. The categories posing challenges include classes A25 and A41 in rectangle A, and classes A9, A14, A34, and A37 in rectangle B. This TCF mirrors

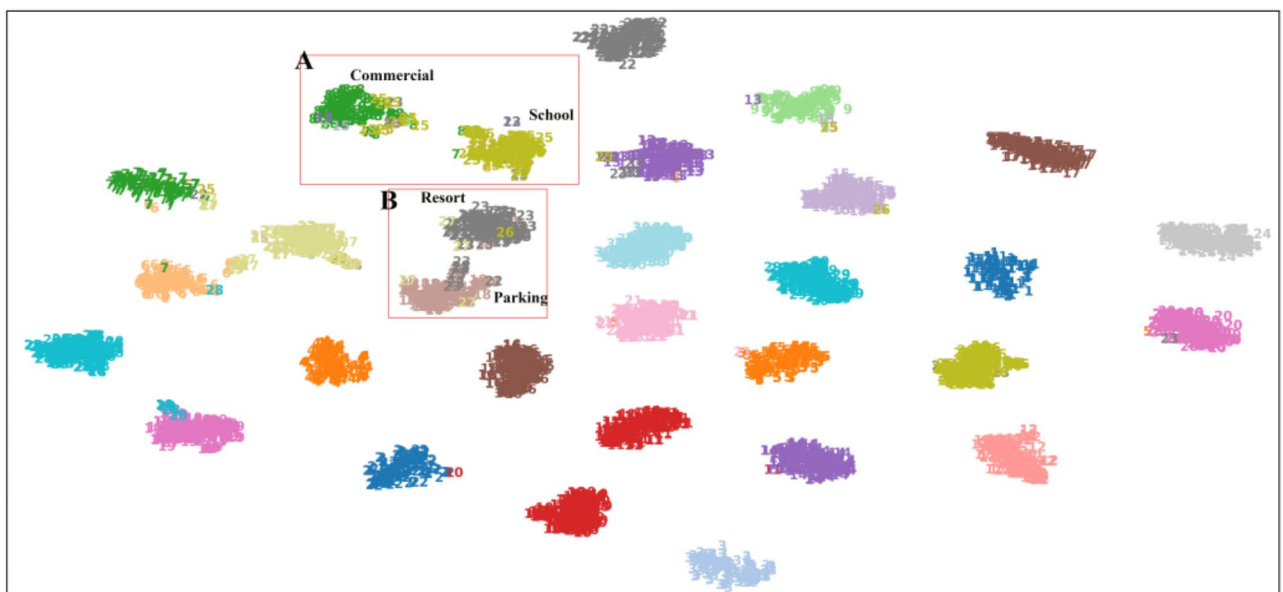


Fig. 9. t-SNE Visualization of RNV-ESE on the AID30 Dataset (TR=20%) (Source: Authors own work by Python version 3.9).

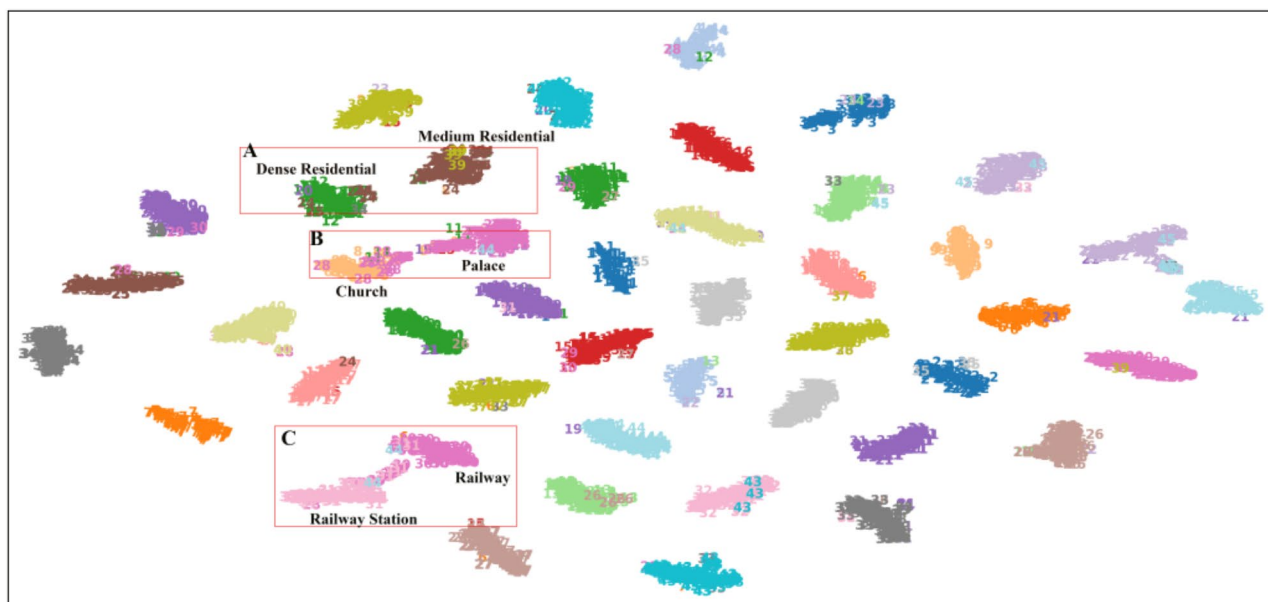


Fig. 10. t-SNE Visualization of RNV-ESE on the NWPU45 Dataset (TR=20%) (Source: Authors own work by Python version 3.9).

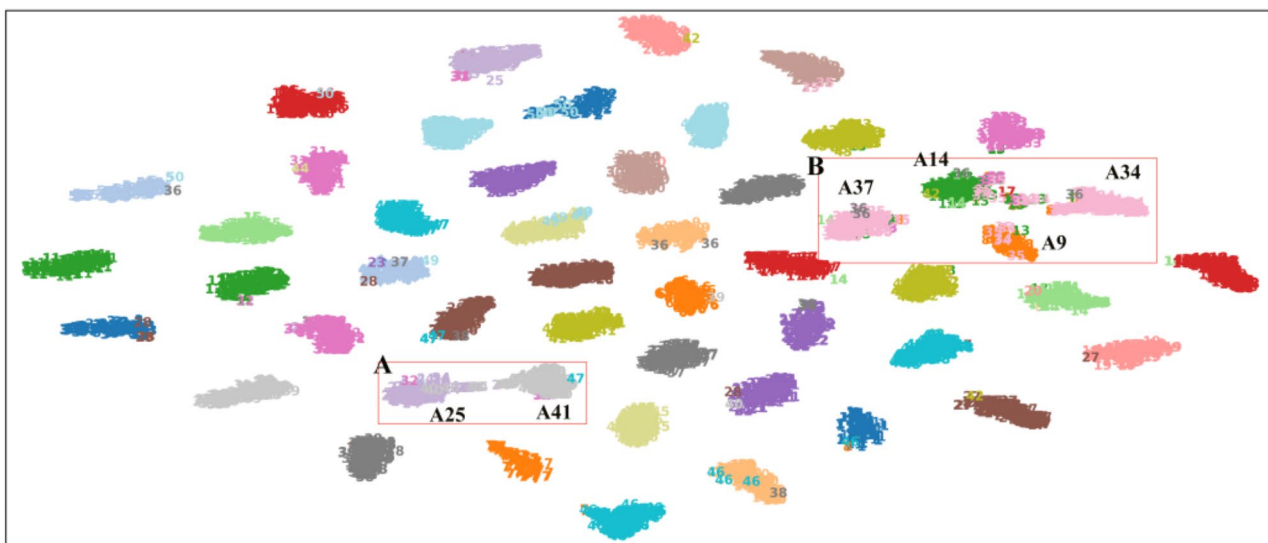


Fig. 11. t-SNE visualization of RNV-ESE on the AFGR50 Dataset (TR=20%) (Source: Authors own work by Python version 3.9).

the difficult categories within AFGR50 that were previously identified by confusion matrices. Furthermore, it substantiates that the misclassification within AFGR50 is more pronounced among all classes compared to those in NWPU45. Essentially, this observation affirms that the TCF for AFGR50 is capable of consistently revealing patterns of misclassification that align with those identified in the confusion matrices. Therefore, we have excluded the TCFs of RST-ESE from our discussion, as their t-SNE results exhibit a similar level of consistency with their confusion matrices.

Contrary to previous studies^{12,46,58–60,62,27–29}, our TCFs display superior clustering results across various datasets, thereby maintaining an impressive performance consistent with the OA results. Consequently, the effectiveness of the RNV-ESE's feature structure is confirmed by all the TCFs.

Computational efficiency

We have compared the computational efficiency of our method with several previous methods, including ViT-Base16⁴⁴, P2FEViT²⁰, and TSTNet²¹. We replicated these methods using only their base models, without any

Method	Params (M)	FLOPs (G)	Inferring time (second)	OA (%)	
				NWPU TR-10%	NWPU TR-20%
ViT-Base16 ⁴⁴	86.6	17.6	49.9 ± 0.01	92.60 ± 0.10	None
P2FEViT ²⁰	> 112.2	> 21.7	72.5 ± 0.06	94.97 ± 0.13	95.74 ± 0.19
TSTNet ²¹	173.0	30.2	149.2 ± 0.08	94.08 ± 0.24	95.70 ± 0.10
RC-B3	12.0	1.8	23.8 ± 0.09	94.69 ± 0.03	96.28 ± 0.05
Swin-T-Tiny	28.3	4.5	38.7 ± 0.02	94.41 ± 0.19	96.22 ± 0.13
N-ViT-S	31.7	5.8	42.4 ± 0.06	94.81 ± 0.05	96.22 ± 0.07
RD-ESE	17.5	2.2	35.2 ± 0.13	95.17 ± 0.05	96.57 ± 0.06
RST-ESE	40.5	6.3	54.1 ± 0.07	95.29 ± 0.04	96.78 ± 0.02
RNV-ESE	43.9	7.6	65.1 ± 0.02	95.34 ± 0.08	96.70 ± 0.04

Table 10. Comparative analysis of computational efficiency across different methods.

Model	Test Item	OA (%)		
		AID30 (TR-20%)	NWPU45 (TR-10%)	AFGR50 (TR-10%)
N-ViT-S	Baseline 1	97.62 ± 0.11	94.81 ± 0.05	92.60 ± 0.17
	Color Jitter (OP = 1.0)	97.64 ± 0.09 ↑ 0.02	94.69 ± 0.05 ↓ 0.12	92.67 ± 0.53 ↑ 0.07
	Horizontal Flip (OP = 1.0)	97.55 ± 0.11 ↓ 0.07	94.74 ± 0.08 ↓ 0.07	92.60 ± 0.30 ↓ 0.00
	Vertical Flip (OP = 1.0)	97.67 ± 0.13 ↑ 0.05	94.67 ± 0.05 ↓ 0.14	92.26 ± 0.41 ↓ 0.34
	Rotation (OP = 1.0)	97.55 ± 0.12 ↓ 0.07	94.61 ± 0.10 ↓ 0.20	92.07 ± 0.24 ↓ 0.53
	Gaussian Blur (OP = 1.0)	97.51 ± 0.03 ↓ 0.11	94.68 ± 0.12 ↓ 0.13	92.71 ± 0.43 ↑ 0.11
	CutMix (OP = 1.0)	97.54 ± 0.15 ↓ 0.08	94.48 ± 0.04 ↓ 0.33	91.89 ± 0.37 ↓ 0.71
	Overdone (All OPs = 1.0)	97.20 ± 0.09 ↓ 0.42	94.14 ± 0.07 ↓ 0.67	91.45 ± 0.25 ↓ 1.15
Swin-T-Tiny	Baseline 2	97.42 ± 0.10	94.41 ± 0.19	91.72 ± 0.34
	Overdone (All OPs = 1.0)	97.23 ± 0.14 ↓ 0.19	93.65 ± 0.10 ↓ 0.76	90.78 ± 0.16 ↓ 0.94
RC-B3	Baseline 3	97.22 ± 0.03	94.69 ± 0.03	91.91 ± 0.34
	Overdone (All OPs = 1.0)	96.85 ± 0.13 ↓ 0.37	94.01 ± 0.11 ↓ 0.68	91.05 ± 0.08 ↓ 0.86

Table 11. OAs (%) from ablation experiment A on the AID30, NWPU45, and AFGR50 datasets.

functional modules proposed in the corresponding literature. This implies that our evaluated inference speeds for these methods are faster than their actual speeds. Given that the number of activations can significantly influence a model's inference speed on GPUs⁷⁰, we used computational efficiency in practical inference as a metric. We conducted experiments with all models by predicting the category label of 25,200 RSI samples at a resolution of 224². The test environment was consistent with the previously introduced one, running on a single RTX-4070Ti GPU. The number of floating-point operations (FLOPs) is measured in giga (G).

As shown in Table 10, our single-model methods demonstrate significant OA improvements when using the TR-20% on NWPU45 as a test bed. Furthermore, our single-model methods have superior inference speeds and fewer parameters. For example, N-ViT-S shows superior OA than the top-ranked TSTNet in the literature, with an 81% and 72% reduction in parameters and inference time, respectively. Moreover, this comparison also reveals that Swin-T is more efficient than ViT-Base16, but only our QA strategy has significantly enhanced the former's capability when compared to TSTNet, which employs dual Swin-T-Base models.

Furthermore, our ESEs have achieved enhanced performance with little sacrifice in model size or inference speed when compared to the top-ranked TSTNet or P2FEViT. For instance, RST-ESE shows superior OA than TSTNet, with a 77% and 64% reduction in parameters and inference time, respectively. Similarly, RNV-ESE significantly outperforms TSTNet, with a 71% and 58% reduction in parameters and inference time, respectively. In particular, RD-ESE outperforms TSTNet, with remarkable reductions of 90% and 76% in parameters and inference time, respectively.

Ablation experiments

To underscore the superiority of our QA method over traditional data augmentation, we performed ablation studies on the AID30, NWPU45, and AFGR50 datasets, respectively. The OA results are presented in Tables 11 and 12. Our method has demonstrated significant advantages, especially when training samples are limited. As such, we employed three challenging TRs as test beds (TR-20% for AID30, TR-10% for NWPU45, and TR-10% for AFGR50).

In Table 11, there are three partitions divided by a bold black line, presenting the ablation results of N-ViT-S, Swin-T-Tiny, or RC-B3, respectively. 'Baselines' refer to the original settings of the proposed QAM during training,

Model	Test Item	OA (%)		
		AID30 (TR-20%)	NWPU45 (TR-10%)	AFGR50 (TR-10%)
N-ViT-S	Baseline 1	97.62 ±0.11	94.81 ±0.05	92.60 ±0.17
	Color Jitter (OP=0.0)	97.54±0.05↓ 0.08	94.67±0.08↓ 0.14	91.98±0.17↓ 0.72
	Horizontal Flip (OP=0.0)	97.51±0.14↓ 0.09	94.77±0.15↓ 0.04	92.50±0.53↓ 0.10
	Vertical Flip (OP=0.0)	97.55±0.17↓ 0.07	94.82±0.11↑ <i>0.01</i>	92.73±0.18↓ <i>0.13</i>
	Rotation (OP=0.0)	97.49±0.06↓ 0.13	94.39±0.07↓ 0.42	90.79±0.44↓ 1.81
	Gaussian Blur (OP=0.0)	97.67±0.10↑ <i>0.05</i>	94.68±0.14↓ 0.13	92.16±0.10↓ 0.44
	CutMix (OP=0.0)	97.02±0.14↓ 0.50	94.30±0.12↓ 0.51	91.99±0.51↓ 0.61
	Underdone (All OPs=0.0)	96.33±0.14 ↓ 1.29	92.73±0.07 ↓ 2.18	85.25±0.21 ↓ 7.35
Swin-T-Tiny	Baseline 2	97.42 ±0.10	94.41 ±0.19	91.72 ±0.34
	Underdone (All OPs=0.0)	95.89±0.16 ↓ 1.53	91.96±0.05 ↓ 2.45	86.03±0.56 ↓ 5.69
RC-B3	Baseline 3	97.22 ±0.03	94.69 ±0.03	91.91 ±0.34
	Underdone (All OPs=0.0)	95.40±0.19 ↓ 1.82	92.06±0.18 ↓ 2.63	84.38±0.80 ↓ 7.53

Table 12. OAs (%) from ablation experiment B on the AID30, NWPU45, and AFGR50 datasets.

while the second to seventh lines of the ‘Test Item’ column correspond to a specific operator being tested. The key difference between our QA and usual data augmentation lies in the fact that the latter persistently activates these operators during model training. As a result, each operator in the first partition has an operation probability (OP) of 100%. In particular, ‘Overdone’ signifies a situation where all operators in the QAM are perpetually active (all OPs=1.0), which aligns more closely with the strategy of qualitative augmentation for model training. In addition, bold or italics values indicate a decrease or increase in OA compared to their baselines, respectively. We have highlighted the crucial information in bolditalics for improved readability. In contrast, Table 13 maintains the same settings as Table 11, except that all OP values are set to 0. These experiments expose the performance degradation across three models when the augmentation strategy is insufficiently robust.

As indicated in Table 11, setting the OP of most QAM operators at 1.0 sequentially reduces the OAs of N-ViT-S on the AFGR50 and NWPU45 datasets. However, the reductions on AID30 are minor. Conversely, setting all QAM operators with a 1.0 OP simultaneously results in substantial reductions for N-ViT-S on all datasets. In comparison, the decreased OAs on NWPU45 and AID30 are smaller and smallest, respectively. Furthermore, we also observed that both Swin-T-Tiny and RC-B3 experience declines in OA, with the OA decline on AFGR50 being the most severe. This observation suggests that RSIs with a higher resolution can mitigate the impact of qualitative regularization on the performance of all models. However, both the CNN and ViT models will still be suboptimal when traditional qualitative regularization is active during training.

As demonstrated in Table 12, disabling a certain GRO within the QAM results in noticeable decreases in OAs for the N-ViT-S model. In particular, the results further reveal that all three models encounter significant OA declines from 1.3 to 7.4% on three datasets when the six GROs in the QAM are all deactivated. Similar to the results in Table 12, the degradation on AID30 is moderate, while that on NWPU45 or AFGR50 is more severe. This deterioration worsens as the resolution of the RSI decreases.

In conclusion, the ablation studies presented in Tables 11 and 12 substantiate that our QA strategy markedly surpasses conventional data augmentation in RSI classification. Our QA approach offers a direct method to unlock the potential of open-source models for efficient and accurate RSI classification.

Conclusions

This study presents a straightforward ensemble method, named ESE, for classifying RSIs with notable accuracy. The proposed method has two unique features. First, we introduce the QA method as a replacement for conventional data augmentation. This approach aims to better incorporate the inherent characteristics of RSIs, thereby significantly enhancing the performance of a single CNN or ViT model. Second, we propose a simple algorithm for generating a lightweight ensemble of dual classifiers, which can efficiently leverage the advantages of both CNNs and ViTs to further improve accuracy.

An evaluation of our method was conducted on three separate RSI datasets. The findings illustrate that our QA-enhanced CNNs and ViTs consistently yield robust and superior accuracy across all datasets. Furthermore, the results confirm that our ESE technique can augment accuracy beyond that of single models. Comprehensive experiments reveal that the exceptional performance of the ESE method primarily stems from the diverse architectures and high precision of its dual classifier components. Sensitivity analyses suggest that the time costs of generating the ESE models can be reduced substantially by 91%, with only a marginal decrease in accuracy of up to 0.05%.

When compared to the other 48 studies from recent literature, our CNN or ViT models exhibit superior OAs, especially when the number of training samples is scarce. Furthermore, our ESE models display more significant improvements in OA, with a minor sacrifice in model size and inference speed. Notably, when compared against the top-ranked method in existing literature, our ESEs not only display an improvement in OA of up to 1.1% on the challenging NWPU45 dataset but also demonstrate a reduction of up to 90% and 74% in parameters and inference time, respectively. In addition, our methods rely solely on open-source models and avoid any intricate alterations to a model structure. Thus, our research offers a user-friendly yet efficient and precise method for RSI

classification, which is particularly advantageous for geoscience researchers with limited experience or time for model reconfiguration.

This study is still in its early stages, and several limitations must be addressed moving forward. First, the proposed method may exhibit sensitivity to hyperparameter settings, which could affect its performance and generalizability. Second, the effectiveness of the approach in other RSI tasks, such as object detection⁷² or knowledge distillation^{73,74}, has yet to be thoroughly evaluated, representing a promising area for future research. Additionally, there is potential for further optimization of the proposed ESE models by reducing redundancy, which could be achieved through a more detailed investigation of various CNN and ViT architectures. The authors plan to expand upon this research in future publications, exploring these avenues and addressing the identified limitations.

Data availability

Readers can access the three datasets within this research at https://github.com/HuaxiangSong/Datasets_Locations (accessed on July 12, 2024).

Received: 12 July 2024; Accepted: 7 February 2025

Published online: 14 February 2025

References

- Ranjan, P. & Girdhar, A. Deep siamese network with handcrafted feature extraction for hyperspectral image classification. *Multimedia Tools Appl.* **83** (1), 2501–2526. <https://doi.org/10.1007/s11042-023-15444-4> (2024).
- Ranjan, P., Girdhar, A. & Kumar, R. A novel spectral-spatial 3D auxiliary conditional GAN integrated convolutional LSTM for hyperspectral image classification. *Earth Sci. Inf.* **17** (6), 5251–5271. <https://doi.org/10.1007/s12145-024-01451-y> (2024).
- Ranjan, P. & Girdhar, A. A comprehensive systematic review of deep learning methods for hyperspectral images classification. *Int. J. Remote Sens.* **43**, 6221–6306. <https://doi.org/10.1080/01431161.2022.2133579> (2022).
- Song, H. A consistent Mistake in Remote sensing images' classification literature. *Intell. Autom. Soft Comput.* **37** (2), 1381–1398. <https://doi.org/10.32604/iasc.2023.039315> (2023).
- Song, H. A leading but simple classification method for remote sensing images. *AETIC* **7** (3), 1–20. <https://doi.org/10.33166/AETIC.2023.03.001> (2023).
- Ao, L. et al. TPNAS: a two-phase evolutionary neural Architecture search for remote sensing image classification. *Remote Sens.* **15**(8), 2212. <https://doi.org/10.3390/rs15082212> (2023).
- Broni-Bediako, C., Murata, Y., Mormille, L. H. B. & Atsumi, M. Searching for CNN Architectures for Remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13. <https://doi.org/10.1109/TGRS.2021.3097938> (2022).
- Shen, J., Cao, B., Zhang, C., Wang, R. & Wang, Q. Remote sensing scene classification based on attention-enabled progressively searching. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13. <https://doi.org/10.1109/TGRS.2022.3186588> (2022).
- Yun, S. et al. CutMix: Regularization strategy to train strong classifiers with localizable features, in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, 6022–6031 (IEEE, 2019). <https://doi.org/10.1109/ICCV.2019.00612>
- Song, H., Wei, C. & Yong, Z. Efficient knowledge distillation for remote sensing image classification: a CNN-based approach. *IJWIS* **20** (2), 129–158. <https://doi.org/10.1108/IJWIS-10-2023-0192> (2024).
- Shi, C., Zhang, X., Wang, T. & Wang, L. A lightweight convolutional neural network based on Hierarchical-wise Convolution Fusion for remote-sensing scene image classification. *Remote Sens.* **14** (13), 3184. <https://doi.org/10.3390/rs14133184> (2022).
- 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 Sens.* **60**, 1–15. <https://doi.org/10.1109/TGRS.2022.3152566> (2022).
- Wang, D., Zhang, J., Du, B., Xia, G. S. & Tao, D. An empirical study of remote sensing pretraining. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–20. <https://doi.org/10.1109/TGRS.2022.3176603> (2023).
- 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.* **18** (11), 1926–1930. <https://doi.org/10.1109/LGRS.2020.3011405> (2021).
- 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 Sens.* **60**, 1–12. <https://doi.org/10.1109/TGRS.2022.3157671> (2022).
- Chen, W. et al. GCSANet: A global context spatial attention Deep Learning Network for Remote sensing scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 1150–1162. <https://doi.org/10.1109/JSTARS.2022.3141826> (2022).
- Shen, J., Yu, T., Yang, H., Wang, R. & Wang, Q. An attention cascade global-local network for remote sensing scene classification. *Remote Sens.* **14** (9), 2042. <https://doi.org/10.3390/rs14092042> (2022).
- Yang, Y. et al. An explainable spatial-frequency multiscale transformer for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–15. <https://doi.org/10.1109/TGRS.2023.3265361> (2023).
- Li, B. et al. Gated recurrent multiattention network for VHR remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13. <https://doi.org/10.1109/TGRS.2021.3093914> (2022).
- Wang, G. et al. P2FEViT: plug-and-play CNN feature embedded hybrid vision transformer for remote sensing image classification. *Remote Sens.* **15** (7), 1773. <https://doi.org/10.3390/rs15071773> (2023).
- Hao, S., Wu, B., Zhao, K., Ye, Y. & Wang, W. Two-Stream Swin Transformer with Differentiable Sobel Operator for Remote sensing image classification. *Remote Sens.* **14** (6), 1507. <https://doi.org/10.3390/rs14061507> (Mar. 2022).
- Hao, S., Li, N. & Ye, Y. Inductive biased swin-transformer with cyclic regressor for remote sensing scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **16**, 6265–6278. <https://doi.org/10.1109/JSTARS.2023.3290676> (2023).
- Iyer, P., Lal, S. & S. A., and Deep learning ensemble method for classification of satellite hyperspectral images. *Remote Sens. Appl. Soc. Environ.* **23**, 100580. <https://doi.org/10.1016/j.rsase.2021.100580> (2021).
- Nalepa, J., Myller, M., Tulczyjew, L. & Kawulok, M. Deep Ensembles for Hyperspectral Image Data Classification and Unmixing. *Remote Sens.* **13**, 20. <https://doi.org/10.3390/rs13204133> (2021).
- Horry, M. J., Chakraborty, S., Pradhan, B., Shulka, N. & Almazroui, M. Two-speed deep-learning ensemble for classification of incremental land-cover satellite image patches. *Earth Syst. Environ.* **7** (2), 525–540. <https://doi.org/10.1007/s41748-023-00343-3> (2023).
- Wu, Z. et al. Deep learning enables satellite-based monitoring of large populations of terrestrial mammals across heterogeneous landscape. *Nat. Commun.* **14** (1), 3072. <https://doi.org/10.1038/s41467-023-38901-y> (2023).
- Cheng, X. & Lei, H. Remote sensing scene image classification based on mmsCNN-HMM with Stacking Ensemble Model. *Remote Sens.* **14**, 4423. <https://doi.org/10.3390/rs14174423> (2022).
- Zhao, Q., Lyu, S., Li, Y., Ma, Y. & Chen, L. Multigranularity multilevel feature ensemble network for remote sensing scene classification. *IEEE Trans. Neural Netw. Learn. Syst.* **34** (5), 2308–2322. <https://doi.org/10.1109/TNNLS.2021.3106391> (May 2023).
- 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 Sens.* **60**, 1–15. <https://doi.org/10.1109/TGRS.2021.3126770> (2022).

30. Li, Z., Chen, G. & Zhang, T. A CNN-transformer hybrid approach for crop classification using multitemporal multisensor images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **13**, 847–858. <https://doi.org/10.1109/JSTARS.2020.2971763> (2020).
31. Zhao, A., Wang, C. & Li, X. A global+ multiscale hybrid network for hyperspectral image classification. *Remote Sens. Lett.* **14** (9), 1002–1010. <https://doi.org/10.1080/2150704X.2023.2258467> (2023).
32. Meoni, G. et al. The OPS-SAT case: a data-centric competition for onboard satellite image classification. *Astrodrum* **8** (4), 507–528. <https://doi.org/10.1007/s42064-023-0196-y> (Dec. 2024).
33. Shendy, R. & Nalepa, J. Few-shot satellite image classification for bringing deep learning on board OPS-SAT. *Expert Syst. Appl.* **251**, 123984. <https://doi.org/10.1016/j.eswa.2024.123984> (2024).
34. Ranjan, P., Gupta, G., Cross-Domain, A. & Semi-supervised zero-shot learning model for the classification of hyperspectral images. *J. Indian Soc. Remote Sens.* **51** (10), 1991–2005. <https://doi.org/10.1007/s12524-023-01734-9> (2023).
35. Dietterich, T. G. Ensemble methods in machine learning, in Multiple Classifier Systems, vol. 1857, in *Lecture Notes in Computer Science*, vol. 1857, 1–15. doi: https://doi.org/10.1007/3-540-45014-9_1 (Springer, 2000).
36. Sesmero, M. P., Ledezma, A. I. & Sanchis, A. Generating ensembles of heterogeneous classifiers using stacked generalization: generating ensembles of heterogeneous classifiers. *WIREs Data Min. Knowl. Discov.* **5**(1), 21–34. <https://doi.org/10.1002/widm.1143> (2015).
37. Zhang, Y., Liu, J. & Shen, W. A review of ensemble learning algorithms used in remote sensing applications. *Appl. Sci.* **12** (17), 8654. <https://doi.org/10.3390/app12178654> (2022).
38. Liu, Z. et al. Swin transformer V2: Scaling up capacity and resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 11999–12009 (IEEE, 2022). <https://doi.org/10.1109/CVPR52688.2022.011170>
39. Li, J. et al. Next-ViT: Next Generation Vision Transformer for Efficient Deployment in Realistic Industrial Scenarios (2022). <https://doi.org/10.48550/arXiv.2207.05501>
40. He, K., Zhang, X., Ren, S., Sun, J. & Recognition, P. IEEE Conference on Computer Vision and Deep Residual Learning for Image Recognition, Las Vegas, 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>
41. Thapa, A., Horanont, T., Neupane, B. & Aryal, J. Deep learning for remote sensing image scene classification: a review and meta-analysis. *Remote Sens.* **15** (19), 4804. <https://doi.org/10.3390/rs15194804> (2023).
42. Shi, C., Zhang, X., Sun, J. & Wang, L. Remote sensing scene image classification based on self-compensating convolution neural network. *Remote Sens.* **14**(3), 545. <https://doi.org/10.3390/rs14030545> (2022).
43. Bai, L. et al. Remote sensing image scene classification using Multiscale Feature Fusion Covariance Network with Octave Convolution. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14. <https://doi.org/10.1109/TGRS.2022.3160492> (2022).
44. Bazi, Y., Bashmal, L., Rahhal, M. M. A., Dayil, R. A. & Ajlan, N. A. Vision transformers for Remote sensing image classification. *Remote Sens.* **13** (3), 516. <https://doi.org/10.3390/rs13030516> (2021).
45. Alhichri, H., Alswayed, A. S., Bazi, Y., Ammour, N. & Alajlan, N. A. Classification of remote sensing images using EfficientNet-B3 CNN model with attention. *IEEE Access* **9**, 14078–14094. <https://doi.org/10.1109/ACCESS.2021.3051085> (2021).
46. Chen, S. B. et al. Remote sensing scene classification via multi-branch local attention network. *IEEE Trans. Image Process.* **31**, 99–109. <https://doi.org/10.1109/TIP.2021.3127851> (2022).
47. Zhang, W., Jiao, L., Liu, F., Liu, J. & Cui, Z. LHNNet: Laplacian convolutional block for remote sensing image scene classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13. <https://doi.org/10.1109/TGRS.2022.3192321> (2022).
48. Meng, Q. et al. Multilayer feature Fusion Network with spatial attention and gated mechanism for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5. <https://doi.org/10.1109/LGRS.2022.3173473> (2022).
49. Tian, T., Li, L., Chen, W. & Zhou, H. SEMSDNet: a multiscale dense network with attention for remote sensing scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 5501–5514. <https://doi.org/10.1109/JSTARS.2021.3074508> (2021).
50. Wan, H. et al. Lightweight channel attention and multiscale feature fusion discrimination for remote sensing scene classification. *IEEE Access* **9**, 94586–94600. <https://doi.org/10.1109/ACCESS.2021.3093308> (2021).
51. Wang, D. & Lan, J. A deformable convolutional neural network with spatial-Channel attention for remote sensing scene classification. *Remote Sens.* **13**, 5076. <https://doi.org/10.3390/rs13245076> (2021).
52. 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 Sens.* **60**, 1–17. <https://doi.org/10.1109/TGRS.2022.3201755> (2022).
53. Tang, X. et al. Attention consistent network for remote sensing scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 2030–2045. <https://doi.org/10.1109/JSTARS.2021.3051569> (2021).
54. Wang, W., Chen, Y. & Ghamisi, P. Transferring CNN with adaptive learning for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–18. <https://doi.org/10.1109/TGRS.2022.3190934> (2022).
55. Xu, K., Huang, H. & Deng, P. Remote sensing image scene classification based on global-local dual-branch structure model. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5. <https://doi.org/10.1109/LGRS.2021.3075712> (2022).
56. Zhang, J., Zhao, H. & Li, J. TRS: Transformers for remote sensing scene classification. *Remote Sens.* **13** (20), 4143. <https://doi.org/10.3390/rs13204143> (2021).
57. Deng, P., Xu, K. & Huang, H. When CNNs meet vision transformer: a joint framework for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5. <https://doi.org/10.1109/LGRS.2021.3109061> (2022).
58. Ma, J. et al. Homo-heterogenous transformer learning framework for RS scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 2223–2239. <https://doi.org/10.1109/JSTARS.2022.3155665> (2022).
59. 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 Sens.* **61**, 1–15. <https://doi.org/10.1109/TGRS.2023.3265346> (2023).
60. Bi, M. et al. Vision transformer with contrastive learning for remote sensing image scene classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **16**, 738–749. <https://doi.org/10.1109/JSTARS.2022.3230835> (2023).
61. Song, H. & Yang, W. GSCCTL: a general semi-supervised scene classification method for remote sensing images based on clustering and transfer learning. *Int. J. Remote Sens.* **43** (15–16), 5976–6000. <https://doi.org/10.1080/01431161.2021.2019851> (2022).
62. Miao, W., Geng, J. & Jiang, W. Multigranularity decoupling network with pseudolabel selection for remote sensing image scene classification. *IEEE Trans. Geosci. Remote Sens.* **61**, 1–13. <https://doi.org/10.1109/TGRS.2023.3244565> (2023).
63. Wang, G., Zhang, N., Liu, W., Chen, H. & Xie, Y. A Multi-level fusion network for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5. <https://doi.org/10.1109/LGRS.2022.3205417> (2022).
64. Wang, D., Zhang, C. & Han, M. MLFC-net: a multi-level feature combination attention model for remote sensing scene classification. *Comput. Geosci.* **160**, 105042. <https://doi.org/10.1016/j.cageo.2022.105042> (Mar. 2022).
65. Guo, W. et al. Remote sensing image scene classification by multiple granularity semantic learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 2546–2562. <https://doi.org/10.1109/JSTARS.2022.3158703> (2022).
66. 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 Sens.* **60**, 1–15. <https://doi.org/10.1109/TGRS.2020.3048024> (2022).
67. Xu, K., Huang, H., Deng, P. & Li, Y. Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing. *IEEE Trans. Neural Netw. Learn. Syst.* **33** (10), 5751–5765. <https://doi.org/10.1109/TNNLS.2021.3071369> (2022).
68. Liu, Z. et al. Swin transformer: hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 9992–10002 (IEEE, 2021). <https://doi.org/10.1109/ICCV48922.2021.00986>

69. Song, H. MBC-Net: long-range enhanced feature fusion for classifying remote sensing images. *IJICC* **17** (1), 181–209. <https://doi.org/10.1108/IJICC-07-2023-0198> (2024).
70. Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K. & Dollar, P. Designing network design spaces, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 10425–10433 (IEEE, 2020). <https://doi.org/10.1109/CVPR42600.2020.01044>
71. Selvaraju, R. R. et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **128** (2), 336–359. <https://doi.org/10.1007/s11263-019-01228-7> (2020).
72. Song, H. et al. QAGA-Net: enhanced vision transformer-based object detection for remote sensing images. *Int. J. Intell. Comput. Cybern.* <https://doi.org/10.1108/IJICC-08-2024-0383> (2024).
73. Song, H. et al. Variance consistency learning: enhancing cross-modal knowledge distillation for remote sensing image classification. *AETiC* **8** (4), 56–76. <https://doi.org/10.33166/AETiC.2024.04.003> (2024).
74. Song, H. et al. ERKT-Net: implementing efficient and robust knowledge distillation for remote sensing image classification. *EAI Endorsed Trans. Ind. Net Intel Syst.* **11** (3). <https://doi.org/10.4108/eetinis.v11i3.4748> (2024).

Acknowledgements

This work was funded by the Hunan Provincial Department of Education's Scientific Research Project (Project No. 24A0482) and the Research Foundation of Hunan University of Arts and Science (Geography Subject [2022] 351).

Author contributions

Investigation, data, and formal analysis: Hanglu Xie, Yingying Duan, Xinyi Xie, Fang Gan, and Wei Wang; methodology and software: Huaxiang Song and Jinling Liu; writing—original draft: Huaxiang Song and Jinling Liu; writing—review and editing: Huaxiang Song. All authors have read and agreed to the published version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-89735-1>.

Correspondence and requests for materials should be addressed to H.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025