*Research Article*

# A Leading but Simple Classification Method for Remote Sensing Images

**Huaxiang Song**

School of Geography Science and Tourism, Hunan University of Arts and Science, China
cn11028719@huas.edu.cn

Received: 15th May 2023; Accepted: 12th June 2023; Published: 1st July 2023

**Abstract: Recently, researchers have proposed a lot of deep convolutional neural network (CNN) approaches with obvious flaws to tackle the difficult semantic classification (SC) task of remote sensing images (RSI). In this paper, the author proposes a simple method that aims to provide a leading but efficient solution by using a lightweight EfficientNet-B0. First, this paper concluded the drawbacks with an analysis of mathematical theory and then proposed a qualitative conclusion on the previous methods' theoretical performance based on theoretical derivation and experiments. Following that, the paper designs a novel method named LS-EfficientNet, consisting only of a single CNN and a concise training algorithm called SC-CNN. Far different from previous complex and hardware-extensive ones, the proposed method mainly focuses on tackling the long-neglected problems, including overfitting, data distribution shift by DA, improper use of training tricks, and other incorrect operations on a pre-trained CNN. Compared to previous studies, the proposed method is easy to reproduce because all the models, training tricks, and hyperparameter settings are open-sourced. Extensive experiments on two benchmark datasets show that the proposed method can easily surpass all the previous state-of-the-art ones, with an outstanding accuracy lead of 0.5% to 1.2% and a remarkable parameter decrease of 78% if compared to the best prior one in 2022. In addition, ablation test results also prove that the proposed effective combination of training tricks, including OLS and CutMix, can clearly boost a CNN's performance for RSI-SC, with an increase in accuracy of 1.0%. All the results reveal that a single lightweight CNN can well tackle the routine task of classifying RSI.**

**Keywords:** ***Deep learning; Image classification; LS-EfficientNet; Remote sensing; SC-CNN algorithm***

## 1. Introduction

Remote sensing (RS) is an important technique for Earth observations, and the imaging picture is big data, containing spatial, spectral, and graphic information. Machine learning (ML) plays a central role in interpreting RSI to meet the domain-specific requirements for real-time and automation. Recently, with the rise of deep learning (DL), deep CNNs have dominated the recognition tasks of RSI. Among all the applications, SC is the foundation of all the others. As a hotspot, researchers have proposed different algorithms to improve the method's performance, though most of them are suboptimal.

At the beginning, CNNs were only used as fixed feature extractors without effective training on RSI datasets [1, 2]. Therefore, the method's performance is poor. Then, with improved training on RSI, the fusion strategy consisting of deep and human-engineered features [3, 4], as well as the one by redesigning loss functions [5, 6], were proposed. These two roadmaps do show a clear advance on the fixed extractor because of the different training. Following that, researchers tend to employ more and more complicated and hardware-extensive technical pipelines to seek better performance [7], though the final benefit is very limited. As the vision transformer (VT) arose [8], more and more methods turned to employing this new architecture without considering the larger parameter size of VT. Based on the rich application scenarios, the larger volume of VT is acceptable for natural images. However, RS applications are basically focusing on the public's welfare, and more importantly, many tasks are using embedded systems, which have strict

Huaxiang Song, "A Leading but Simple Classification Method for Remote Sensing Images", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 1-20, Vol. 7, No. 3, 1st July 2023, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2023.03.001, Available: http://aetic.theiaer.org/archive/v7/v7n3/p1.html.

hardware limitations. Hence, compared to the VT, a CNN with fewer parameters is optimal if its performance can surpass the VT.

Generally, CNNs are more discriminative if their representation consists of more invariant features. Currently, CNN-based techniques for RSI-SC commonly employ pre-trained models on ImageNet-1K from natural images. The inherent difference in RSI requires a fine-tuning process for feature extraction. However, according to the author's knowledge, all the previous methods went in the wrong direction. First, some methods ignore or inappropriately implement a fine-tuning process. Natural images have common invariant features with RSI, although an inherent difference also exists. But many previous methods employed improper training strategies, e.g., an oversized learning rate (LR). It has made the pre-trained model totally discard valuable general features and overfit on the small RSI dataset. Second, some previous methods made modifications to a pre-trained model's structure without retraining it on ImageNet-1K. It will face performance degradation or overfitting problems too. Third, currently, many training tricks developed on ImageNet-1K can greatly boost CNN's performance. But most of the previous methods proposed for RSI-SC simply copied the training trick without rechecking its usability. Fourth, data augmentation (DA) does improve a model's performance with finite training samples. But datasets processed by DA have a clear shift in data distribution compared to the original ones. In other words, a model will achieve suboptimal performance if the shifting problem is not well handled [9]. However, to the author's best knowledge, the problem has been totally ignored in previous studies. More importantly, all the arbitrary training algorithms that have long existed in the literature have made us encounter a dilemma. Too much randomness is correlating with the findings of previous studies. It is hard to tell what is really meaningful without an appropriate training procedure.

To solve this problem, at the beginning of this study, the author makes a deep analysis of all the previous methods and proves in mathematical theory that many of the complicated and hardware-extensive methods are much more difficult to achieve better performance because of improper algorithm choices. In the subsequent sections, the study proposes a novel CNN-based method for RSI-SC. Different from previous ones, the proposed method, called the leading and simple EfficientNet (LS-EfficientNet), only has a single EfficientNet-B0 with a much smaller number of 5.3 million (M) parameters [10]. The method still employs transfer learning but handles the above problems well through a concise training algorithm. Extensive experiments on two benchmark datasets show that LS-EfficientNet outperforms all the previous methods remarkably. The author summarizes the study's contributions as below:

First, the author proposes a leading but efficient method for RSI-SC. It outperforms all the previous CNN-based methods before 2023 with the fewest parameters but excludes complicated architecture modifications. It can be easy to reproduce because the model is off-the-shelf.

Second, the author rechecked the availability of two training tricks developed on ImageNet-1K and redesigned their usage to boost the CNN's performance for RSI-SC. The rebuilt tricks can lift the model's accuracy by approximately 1.0%, and the hyperparameter settings are open source.

Third, the study reveals some fundamental mistakes in the CNN-based methods for RSI-SC. Taking all these findings together, we can see that all the previous CNN-based methods may have had suboptimal performance compared to their potential capabilities.

## 2. Related Works

With add-in attention modules, a CNN model can obtain better performance on ImageNet-1K. Based on this effective but hardware-cheap technique, researchers have proposed different methods by adding attention modules to pre-trained CNNs for RSI-SC. E.g., Tong *et al*. [11], Guo *et al*. [12], and Guo *et al*. [13] designed spatial or channel attention modules to boost a single pre-trained CNN's performance. Li *et al*. [14] employed attention maps to guide a pre-trained CNN to learn so-called discriminative representation. Alhichri *et al*. [15] also proposed another method by using a pre-trained EfficientNet-B3 model with built-in attention modules. With the help of the attention mechanism, all these methods using a single CNN show obvious improvements over the ones using feature fusion or modified loss functions.

Following that, more and more complicated and hardware-extensive methods emerge, though the advance is not obvious. E.g., Tang *et al*. [16] employed two CNNs with attention modules in a parallel pipeline and trained the models using a concatenate loss function. Sun *et al*. [17] designed a cascaded

method consisting of a CNN, a gated bidirectional network, and a classifying module. Zhang *et al*. [18] combined a CNN with a CapsNet in a series-connected method. Li *et al*. [19] proposed a multi-process method by first extracting deep features with a pre-trained CNN, then refining the extracted features with attention modules, and finally feeding the refined features to deep-gated recurrent units. Chen *et al*. [20] fused five so-called context modeling blocks with a DesenNet-121. Putting the limited improvements aside, we can find that all these methods consist of multiple models or complex modifications to the model's architecture. It has an obvious larger hardware budget or is hard to reproduce due to a lack of open source.

As another effective and well-known technique, an ensemble of CNNs is also tested for RSI-SC. E.g., Minetto *et al*. [21] proposed a CNN ensemble consisting of twelve independent CNNs. Zhao *et al*. [22] proposed a compact ensemble consisting of a CNN backbone with multiple attention module branches. These two methods both have different classifiers in the ensemble, but the individual members of the models or modules are improperly trained on RSI or not retrained on ImageNet-1K. In theory, an ensemble is more effective only when each individual classifier in the ensemble is accurate and diverse. Therefore, the two ensembles' performances have shown a temporary lead over the single CNN-based method but are still suboptimal.

Speaking from another viewpoint, the training algorithm is also crucial for CNN's performance. E.g., He *et al*. [23] conclude that a set of training tricks, including label smoothing [24], Mixup [25], and so on, are very meaningful to the CNN's performance. These tricks were employed in [11, 13, 20] and demonstrated to be effective for RSI-SC. However, in the previous methods, the trick's usage was simply copied from ImageNet-1K without any modification. In fact, label smoothing is commonly used in CNN's training as regularization, but it sets an equal value for all subclasses, with the difference in similarity ignored. Despite the smaller category similarity in natural images, an alternative online label smoothing (OLS) technique proposed by Zhang *et al*. [26] clearly improved CNN's performance on ImageNet-1K by dynamically updating the subclass soft label. Similarly, as regularization in CNN's training, Mixup overlays different image patches on a training sample unnaturally without moving the original overlapped part of the original image. Yun *et al*. [27] proposed an alternative algorithm called CutMix that shows improved training efficiency. Hence, taking the inherent difference between RSI and the advance shown by updated training tricks into account, it is reasonable and meaningful to make a thorough analysis of the usability of tricks before applying them to RSI-SC.

In addition, CNN's training process commonly uses DA to boost a model's performance, but it also comes with a data distribution shift to the original datasets. Touvron *et al*. [9] demonstrate a fine-tuning or empirically correcting solution for ImageNet-1K. This compromised method is considerable for large-scale datasets but costly for RSI. Tan *et al*. [28] also propose a progressive learning method by employing more intensive regularization as the training goes deeper. Similarly, this optional plan is practicable if computing resources are sufficient. Previous studies for RSI-SC commonly employed DA in training; however, no work has mentioned handling the shifting problem. E.g., Zhang *et al*. [29] proposed an optimized training strategy by using multi-size images with triplet loss for RSI-SC, but with the shifting problem uncorrected, the method's accuracy showed litter improvement. Therefore, based on the above problems, the authors propose a novel training algorithm named shifting corrected CNN (SC-CNN), and its uniqueness is presented as follows:

First, the training procedure consists of a continuous pipeline but can be viewed as two steps based on the different DAs and regularizations in training. Second, both steps have similar routine geometric transformations but with different regularizations. Third, based on the inherent difference in RSI, the usage of training tricks in SC-CNN is rechecked first and then set for different hyperparameters compared to the ones developed on ImageNet-1K. Last, SC-CNN does not include the ideas proposed in [9, 28], and it is totally different from the other previous methods.

## 3. Methodologies

### 3.1. Mathematical Basis

Let $x_i$ be an RS image and $y_i$ be its category label, then a RSI set $S$ can be described as the form:

$$S = \{(x_1, y_1), \cdots, (x_n, y_n)\} \quad (1)$$

Using this notation, the relationship between $x_i$ and $y_i$ can be defined as follows:

$$y = f(x) \tag{2}$$

Since different CNN architectures can be used in the classification, we can treat each CNN model as a hypothesis of the true function $f$. In training, we employ the back-propagation algorithm to minimize the loss of a CNN prediction. In such cases, we can treat the training algorithm as a search program for the optimal solution to a certain hypothesis.

In ML, we commonly use finite iteration steps to fit the training dataset. As the dataset gets much larger in DL, we employ a mini-batch of samples and take the batch's mean value to optimize the training model. Therefore, unlike the shallow model used in ML, we always achieve many local optimal solutions for CNNs due to a lack of exhaustion.

Currently, deep CNN architectures commonly consist of a cascade of convolution layers. In each layer, it has different convolution kernels, and the assigned values for these kernels are the main parameters of a CNN. Let us suppose that the value of the kernel parameter, in the simplest case, can only be set to 0 or 1. If a layer has a certain number of $n$ kernels, then the total number of all kernel states $N$ can be described as follows:

$$N = 2^n \tag{3}$$

Let us suppose that a CNN has a number of $m$ layers, and in the simplest case, all the layers have the same structure. Then, the $N$ in this CNN can be described as follows:

$$N = 2^{n \times m} \tag{4}$$

In fact, at each iteration step, we can only change all the parameter states of a CNN for a certain permutation. Hence, we can see that the searching space for the training algorithm is linearly correlated with the $N$ in a CNN. As a CNN's size grows, e.g., the number of layers gets larger, then, according to Eq. (4), we can see that the searching space for the training algorithm to get local optimal solutions will show an exponential growth. To tackle this problem, we generally have two choices. First, we choose a pre-trained model, and then we will get a good starting point for searching. Second, we choose a larger LR, and then we will get a fast but salutatory searching process. It may make the search miss out on lots of optimal solutions. Recently, attention modules have been widely used for CNNs. Taking the famous channel attention, i.e., squeeze-and-excitation (SE) module [30], as an example, it works in this way.

Let $U_o \in \mathbb{R}^{C \times H \times W}$ be the original feature map of a certain layer, with a number of $C$ channels, a height of $H$, and a width of $W$. Let $U_t \in \mathbb{R}^{C \times H \times W}$ be the transformed feature map by SE modules. Let $F_{sq}$ be the squeeze operation and $F_{ex}$ be the excitation operation. Then, the SE working pipeline can be described as follows:

$$U_t = F_{ex}\left(F_{sq}(U_o)\right) \times U_o \tag{5}$$

In detail, the $F_{sq}$ makes a squeeze on the $U_o$, and its output $U' \in \mathbb{R}^{C \times 1 \times 1}$ will have the same height and width as $1 \times 1$. Afterwards, the $F_{ex}$ makes an excitation to the $U'$, and its output, $U'' \in \mathbb{R}^{C \times H \times W}$, will have the same shape as $U_o$. In training, the $U''$ will be a weighted feature map with larger and smaller values. We can see that, according to Eq. (5), the $U_0$ will also be weighted by the multiplication. In the back-propagation process, a value of features close to zero means less important information. In other words, the SE module makes the input feature map partially meaningful for a CNN. Under this condition, according to Eq. (4), we can see that the searching space of a CNN with built-in attention modules is pre-marked through pre-training. Therefore, with finite training steps, a CNN with attention modules can be more discriminative only if the dataset has significant features.

Based on this explanation, however, if the attention modules have no pre-training and are initialized at random, we can see that the searching space will be the same as the CNN without attention. Hence, looking at previous studies in [11–15], we can see that the methods without pre-training on ImageNet-1K will probably be suboptimal. Taking previous studies [16–20] into account, according to Eq. (4), we can see that these methods commonly get a much larger searching space when multiple models are combined in series. More importantly, these methods commonly train the combined models with a single loss function simultaneously, resulting in a poor probability of getting optimal solutions. Hence, based on mathematical theory, we can find that these hardware-extensive strategies are unnecessary if the single model works well.

CNN's classification ability relies on the various patterns in datasets. If we choose a model, then its capacity for patterns is fixed. Let the probability of a sample belonging to a subclass be $P \in [0,1]$, and then we can describe the model's prediction P as follows:

$$P = \sum_{i=1}^{n} W_i \times Pat_i \tag{6}$$

where $Pat_i$ denotes all the learned patterns contained in a model and $W_i \in [0,1]$ is the weight of a pattern to determine its contribution to a certain subclass.

CNN recognizes different patterns through a combination of parameters. That is, the parameters of the convolution kernels control the extraction of features. Let us use the same $Pat_i$ as Eq. (6), and then we can describe the feature extraction as follows:

$$Pat_i = \sum_{j=1}^{m} F_{conv(Output_{j-1})} \tag{7}$$

where $Output_j$ denotes the previous layer's output, $F_{conv}$ denotes the convolutional operation of the current layer, and $j$ corresponds to the number of layers.

Let us use the same $P$, $F_{conv}$, and $Output_j$ in Eqs. (6) and (7), then we can describe the changing process of a model's prediction during training as follows:

$$P = \sum_{i=1}^{n} \sum_{j=1}^{m} W_i \times F_{conv(Output_{j-1})} \tag{8}$$

As the training algorithm updates the kernel's parameter, CNNs can extract more local invariant features from an RSI dataset and gradually replace the ones learned through pre-training on ImageNet-1K. As the fitting process goes further, a CNN's prediction is more accurate only if the local feature is more general and discriminative. However, if the training and testing datasets have very different distributions, overfitting occurs.

Currently, CNN-based methods completely leverage the pre-trained weights of ImageNet-1K to conduct RSI-SC. The reason lies in two facts. First, the feature of large-scale datasets is more general. Second, the RS domain lacks large-scale datasets. Compared to the one million samples in ImageNet-1K, the two benchmark RSI datasets, including the Aerial Image dataset (AID) and the Northwestern Polytechnic University Remote Sensing Image Scene Classification 45 dataset (NWPU), only have 10,000 and 31,500 samples, respectively [31]. Therefore, a good training algorithm should avoid overfitting on the smaller RSI datasets. In addition, some viewpoints believe that the feature from ImageNet-1K has a great domain gap with RSI. Nonetheless, we can easily evaluate the pre-trained model's ability for RSI-SC with a fast test. The experiment just includes a layer-frozen operation for a CNN model of EfficientNet-B0, including all convolution layers frozen except the classifier. Then it trains the model on the RIS datasets at a fixed training ratio (TR) to find OA results. As shown in Figure 1, the evaluation results on AID and NWPU, including four different TRs, prove that the pre-trained model on ImageNet-1K has an acceptable accuracy of approximately 75% for RSI-SC.

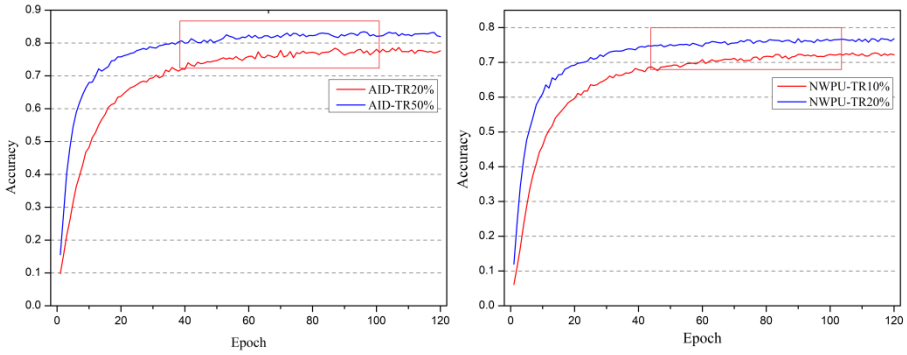



**Figure 1.** Layer-frozen experiments on AID and NWPU.

Therefore, based on all the above explanations in mathematical theory, the author designed a concise and simple method consisting of a lightweight EfficientNet-B0 model with built-in attention modules and two modified training tricks as regularization in training to avoid overfitting problems.

### 3.2. Method's Framework

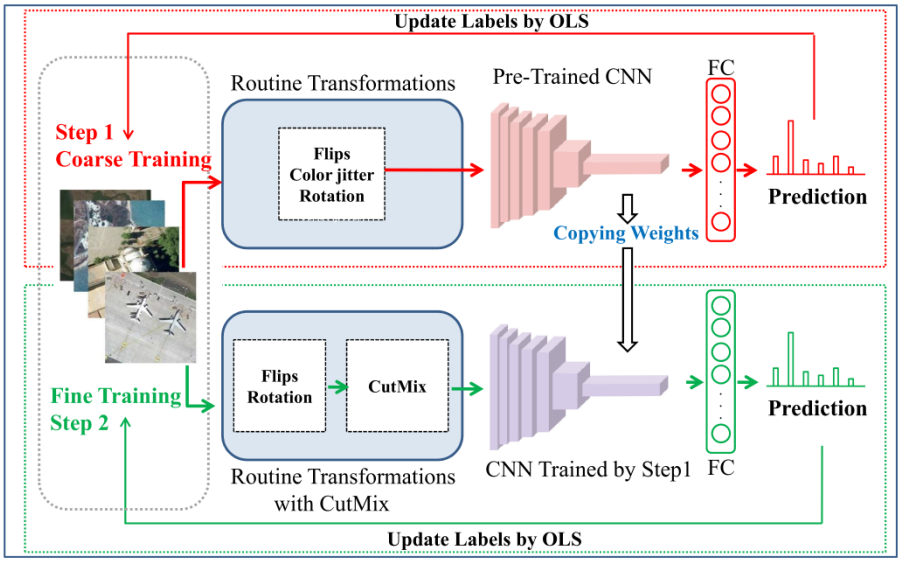



**Figure 2.** The proposed method's framework

The proposed method's framework is illustrated in Figure 2. The algorithm's whole pipeline, as shown in Figure 2, consists of continuous procedures of 300 training epochs in total, but can be viewed as two successive steps according to the different DA and regularizations used in training. In detail, the training process starts with the red arrows corresponding to Step 1, called coarse training, and then, at epoch 61, the training procedures start with the green arrows corresponding to Step 2, named fine training. In Step 1, the pre-trained EfficientNet-B0 is coarsely trained on RSI datasets for 60 epochs, and successively, in Step 2, the model inherits the weights related to the best OA of Step 1 and keeps training on RSI datasets for another 240 epochs. The biggest difference between Step 1 and Step 2 is the training epochs, DA, and regularizations, which are presented in subsequent sections.

### 3.3. DA Strategies

The proposed method employs four kinds of routine transformations in a cascaded combination, and all the transformations are implemented via the PyTorch libraries. In Step 1, it consists of the color jitter, horizontal flip, vertical flip, and rotation in turn. In Step 2, it only consists of the horizontal flips, vertical flips, and rotation.

### 3.4. Regularization

The proposed method employs the modified OLS and modified CutMix as regularization, while the former exists throughout the whole pipeline but the latter is only in Step 2.

#### 3.4.1. OLS Settings

The OLS quantifies the difference in similarity among subclasses by dynamically updating the soft label in training. The algorithm, as shown in [26], initializes the learnable soft labels at zero. Hence, the training loss still needs a traditional hard label to improve the speed of convergence. Let $Loss$ denote the final loss in training, then it can be described as follows:

$$Loss = (1 - \alpha) \times Loss_{hard} + \alpha \times Loss_{soft} \tag{9}$$

where $\alpha$ is the hyperparameter to balance the hard and soft losses. Zhang *et al*. [26] proposed an empirical value of 0.5 for $\alpha$. Here, taking the larger similarity in categories of RSI and also based on extensive experiments, the author sets $\alpha$ at an empirical value of 0.9.

#### 3.4.2. CutMix Settings

The CutMix algorithm, as shown in [27], first randomly cuts an A-class image patch and then replaces an equal area of another B-class image with the cut patch. Let $Label_{cm}$ denote the cut-and-mixed image's label, then it can be described as follows:

$$Label_{cm} = \begin{cases} (1-\beta) \times Label_A + \beta \times Label_B, & if\ Prob \geq \gamma \\ \beta \times Label_B\ , & else \end{cases} \quad (10)$$

where $\beta$ is a hyperparameter equal to the ratio of the cropped area to the original one, and $\gamma$ is another hyperparameter that controls the occurrence probability of a cut-and-mix operation. Yun *et al*., as shown in [27], proposed the beta distribution function to obtain the value of $\beta$ and an empirical value of 0.9 for $\gamma$. In this paper, however, the author uses the same method to obtain β but sets the value of $\gamma$ at 0.1. The reason can be simply explained as follows:

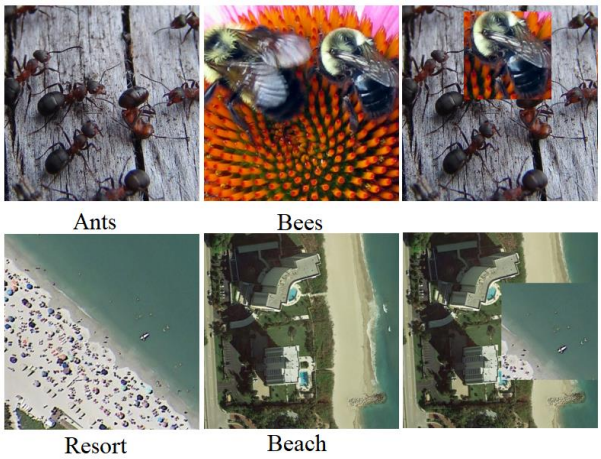



**Figure 3:** The cut-and-mixed samples

The cut-and-mixed samples, as shown in Figure 3, show a larger difference between RSI and ImageNet-1K. Looking at the top of Figure 3, the algorithm randomly cuts a bee-class patch and mixes it with an image of ants, and then, according to Eq. (10), the label of the cut-and-mixed image is 0.2 bees and 0.8 ants, which may be fine. Speaking of the bottom of Figure 3, however, the confidence level is obviously lower if the label of the cut-and-mixed image is 0.2 of beach. Extensive experiments prove that the model will be suboptimal if we use a larger occurrence probability for the cut-and-mix operation.

## 4. Experiments

### 4.1. Model Architecture

The proposed method employs the EfficientNet-B0 as the single model, which is the smallest of the EfficientNets with significantly fewer parameters (5.3 M). The architecture of EfficientNet-B0, as shown in [10], has built-in SE blocks. In this paper, the model's default settings, including architecture, dropout, and stochastic depth, are unchanged. Given the pre-trained EfficientNet-B0, only its last classifier is reset according to the subclass number of the RSI datasets.

### 4.2. Training Algorithm

**Algorithm 1.** The procedures of the proposed SC-CNN

**Definition**: training dataset $S_{train} = \{(x_i, y_i)\}$, testing dataset $S_{test} = \{(x_i, y_i)\}$, EfficientNet-B0 model $f$, transformations in Step1 $Trs_1$, transformations in Step2 $Trs_2$, OLS algorithm $f_{OLS}$, CutMix transformation $f_{cm}$, cross-entropy error function $f_{cee}$, model's prediction accuracy Acc, Acc dictionary Results,

Initializing: number of training samples $N_{train}$, number of testing samples $N_{test}$

Step 1:

For Epoch=1, 2, . . . , 60 do

  For iteration = 1 to $\left(\frac{N_{train}}{30} + 1\right)$ do

    Sampling a batch of samples $B \in S_{train}$, inputting to $f$

    Predicting probabilities $\hat{y}_i = f\left(Trs_1(x_i)\right)$

    Calculating loss $loss = \left(\hat{y}_i - f_{OLS}(y_i)\right)$

    Updating parameters through back propagating

  End For

  $Acc = (f(x_i) == y_i),\ x_i, y_i \in S_{test}$

9 If Acc is the best then
10 Save Acc in Results
11 End For
Step 2:
12 For Epoch=61, 62, . . . , 300 do
13 For iteration = 1 to $\left(\frac{N_{train}}{30}+1\right)$ do
14 Sampling a batch of samples $B \in S_{train}$, inputting to f
15 Predicting probabilities $\hat{y}_i = f\left(f_{cm}(Trs_1(x_i))\right)$
16 Calculating loss $loss = \left(\hat{y}_i - f_{OLS}(f_{cm}(y_i))\right)$
17 Updating parameters through back propagating
18 End For
19 $Acc = (f(x_i) == y_i),\ x_i, y_i \in S_{test}$
20 If Acc is the best then
21 Save Acc in Results
22 End For
23 Return Results

The SC-CNN algorithm, as shown in Algorithm 1, is a typical transfer learning strategy written in Python. Given the same resolution of $256^2$ for training and testing, the total number of training epochs in Step 1 is 60, while the one in Step 2 is 240. Note that the setting epochs are empiric values according to the evaluation result in Figure 1.

The method employs cross-entropy as the object function. The error-back-propagation algorithm is the Adam-W [32], with a weight decay of 1E-06. In Steps 1 and 2, the initial learning rate is both 1E-04 with cosine decay, and for cosine decay settings, the maximum number of iterations is 60 and 240, respectively. The training mini-batch is fixed at 30 for all datasets.

### 4.3. Dataset and Division


**Figure 4:** Typical samples in AID


**Figure 5:** Typical samples in NWPU

This study employs two RSI datasets as benchmarks, including AID and NWPU, and the samples from each category are shown in Figures. 4 and 5. More details about these two datasets can be found in [31]. To get a fair comparison, the TRs are the same as in previous studies, including 20% and 50% for AID but 10% and 20% for NWPU. All the training and testing subsets are chosen at random.

### 4.4. Evaluation Criteria

This study employs the OA and confusion matrix [31] as criteria for performance evaluation. Let $N_c$ be the total number of accurately classified samples and $N_t$ be the total number of tested samples, the OA can be described as follows:

$$OA = \frac{N_c}{N_t} \tag{11}$$

### 4.5. Hardware and Software Environments

The experiments were performed on four personal computers equipped with a single RTX 2060 GPU. PyTorch 1.11.0 is installed on Windows 10. All the experimental results were averaged over five runs.

## 5. Results

### 5.1. Fitting Curves

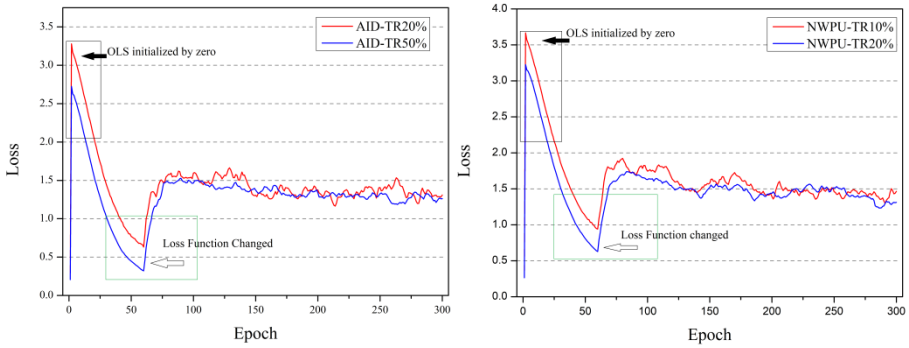



**Figure 6:** Training loss curves for AID and NWPU.

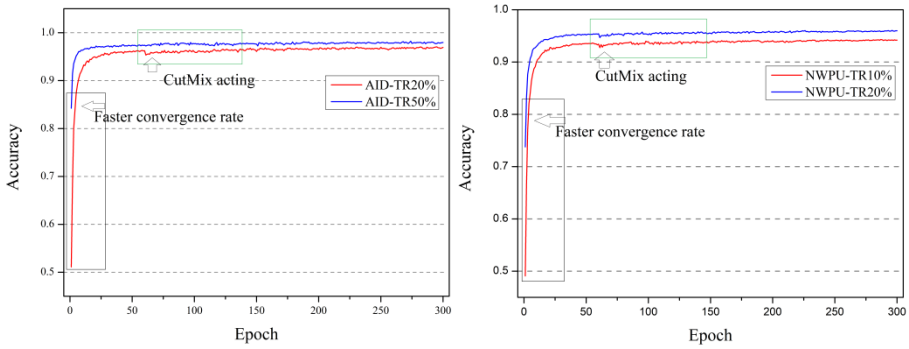



**Figure 7:** Testing accuracy curves for AID and NWPU.

The model's fitting curves on AID and NWPU are shown in Figures. 6 and 7, in which the former is training loss and the latter is testing accuracy.

The model's loss curves, as shown in Figure 6, both show a fast decline in Step 1 but go oscillatory in Step 2 (marked with green rectangles). In Step 1, the label function is Eq. (9), but in Step 2, the function is Eq. (10). Therefore, the loss function changes as the number of training epochs surpasses 60. At the first several epochs, we can see that the loss curves show a fast increase from small values but then decrease clearly in the subsequent epochs (marked with green rectangles). The OLS algorithm, as mentioned before, initializes its soft labels for each category with zero and then dynamically updates the labels as training goes deeper. Hence, with these rapidly declining losses, we can see that the soft label generated by OLS is adaptive to different datasets.

Looking at the model's accuracy curves in Figure 7, we can see that the model shows rapid convergence rates since the first training epochs (marked with black rectangles), but its accuracy presents

a slight decrease at epoch 60 (marked with green rectangles); afterwards, the accuracy curves both climb higher in the following epochs. The cut-and-mixed samples, as explained in Figure 3, have two subclass labels, but the model's prediction may not give larger probabilities for these two subclasses due to the similarity in categories of RSI. Therefore, we can still find notable yields in accuracy, though the loss curves have obviously rebounded. These results prove that, with CutMix as regularization in Step 2, the model has been forced to learn more discriminative features when the cut-and-mixed patch disturbs the image's original label. In addition, we can also see that the pre-trained model on ImageNet-1k can achieve very fast fittings even during the first several epochs; meanwhile, the TRs with more samples can make the model gain a more rapid convergence rate. Hence, these results also reveal that a deep CNN model pre-trained on ImageNet-1K is easy to overfit on the RSI datasets, though many researchers pay more attention to the domain gap between natural images and RSI.

Nonetheless, these results also prove the hypothesis in Eqs. (6) to (8). First, as training goes deeper, the CNN will learn more local features, but the model's performance may be suboptimal due to the data distribution gap between training and testing sets. Second, overfitting is easy to emerge for these RSI datasets, though we used a smaller LR of 1E-04 and only trained for a short period of 60 epochs.

### 5.2. OA Results

**Table 1.** OA (%) comparison of different methods on AID

| Methods | Technical Route | Base model | | TR | |
|---|---|---|---|---|---|
| | | Architecture | Params | 20% | 50 (%) |
| Chaib *et al.* [2] | Feature fusion | Three VGGNet-16 | 415 M | None | 89.71 ±0.33 |
| Liu *et al.* [3] | | Three VGGNet-16 | 415 M | None | 96.37 ± 0.30 |
| | | Three GoogLeNet | 39 M | None | 94.12 ± 0.32 |
| Liu *et al.* [4] | | Single GoogLeNet | 13 M | None | 97.24 ± 0.32 |
| Cheng *et al* [5] | | Single VGGNet-16 | 138 M | 90.82 ± 0.16 | 91.89 ± 0.22 |
| Bazi *et al.* [6] | Loss modification | Single EfficientNet-B0 | 5.3 M | 93.69 ± 0.11 | 96.17 ± 0.16 |
| | | Single EfficientNet-B3 | 12.2 M | 94.19 ± 0.15 | 96.56 ± 0.14 |
| Xie *et al.* [7] | Architecture fine-tune | Single VGGNet-16 | 138 M | 93.60 ± 0.12 | 96.66 ± 0.11 |
| Guo *et al.* [11] | Attention module add-in | Partial VGGNet-16 | 57 M | 95.02 ± 0.28 | 96.66 ± 0.19 |
| Tong *et al.* [12] | | Single DensnNet-121 | 8.3 M | 95.73 ± 0.22 | 97.16 ± 0.26 |
| Alhichri *et al.* [14] | | Single EfficientNet-B3 | 12.2 M | 94.45 ± 0.76 | 96.56 ± 0.12 |
| Tang *et al.* [17] | | Two VGGNet-16 | 276 M | 93.33 ± 0.29 | 95.38 ± 0.29 |
| Li *et al.* [18] | | A ResNet-101 with GRUs | 54.1 M | 96.19 ± 0.48 | 97.84 ± 0.39 |
| Sun *et al.* [16] | Multiple models | A VGGNet-16 with two self-designed modules | 12.2 M | 92.20 ± 0.23 | 95.48 ± 0.12 |
| Zhang *et al.*[19] | | An InceptionV3 with CapsNets | None | 93.79 ± 0.13 | 96.32 ± 0.12 |
| Chen *et al.* [21] | Attention module add-in; Multiple models | A DensnNet-121 with multiple self-designed blocks | None | 95.96 ± 0.38 | 97.53 ± 0.32 |
| Zhao *et al.* [22] | A CNN ensemble | A DensnNet-121 with four self-designed branches | 23.9 M | 96.39 ± 0.21 | 98.40 ± 0.23 |
| [this work] | Single CNN | EfficeintNet-B0 | 5.3 M | 96.86 ± 0. 07 | 98.05 ± 0.06 |

To verify the method's effectiveness, the author compares 21 CNN-based methods in previous literature. The presented data includes the method's OAs, the base model's architectures, and parameter sizes. As a fair comparison, the TR is the same, if not specifically stated. Note that most of the previous methods modified the model's architecture or employed multiple models. Hence, the real parameter sizes of these methods should be inconceivably larger. The comparable results for AID are shown in Table 1, while those for NWPU45 are shown in Table 2. Note that "None" means that no relevant results are presented in the literature.

As shown in Table 1, compared to all the previous state-of-the-art CNN methods, the author's method on AID easily outperforms with an outstanding lead on OA; meanwhile, it is undoubtedly more lightweight with the fewest parameters. Compared to all the popular strategies in detail, the author's lead

over the best of feature fusion is 0.81% [4], though the compared method used a smaller testing ratio of 20%; the lead over the best of attention module add-in [18] is 0.2% to 0.6% with a clearly decrease of 91.3% for parameters; the lead over the best of multiple models [21] is 0.5% to 0.9%, though the compared method's parameter size is not presented but undoubtedly be more huge. Besides, the author's lead over the CNN ensemble [22] is 0.5% at a TR of 20% and -0.3% at a TR of 50%. Technically speaking, based on Eq. (11), we can find that the TR of 50% is smaller. Therefore, taking the wrongly labeled samples by humans into account, the author argues that the method's performance evaluation is more persuasive with a larger number of testing samples. Nonetheless, these results show that, compared to the author's lightweight one, all the previous methods do not achieve outstanding improvements on AID, even though more handcrafted features, more parameters, and multiple models are used.

**Table 2.** OA (%) comparison of different methods on NWPU

| Methods | Technical Route | Base model | | TR | |
|---|---|---|---|---|---|
| | | Architecture | Params | 10% | 20 (%) |
| Liu *et al*. [3] | Feature fusion | Three VGGNet-16 | 415 M | None | 93.27 ± 0.17 |
| | | Three GoogLeNet | 39 M | None | 88.43 ± 0.18 |
| Cheng *et al*. [5] | | Single VGGNet-16 | 138 M | 89.22 ± 0.50 | 91.89 ± 0.22 |
| | | GoogLeNet | 13 M | 86.89 ± 0.10 | 90.49 ± 0.15 |
| Bazi *et al*. [6] | Loss modification | EfficeintNet-B0 | 5.3 M | 89.96 ± 0.27 | None |
| | | EfficeintNet-B3 | 12.2 M | 91.08 ± 0.14 | None |
| Xie *et al*. [7] | Architecture fine-tune | VGGNet-16 | 138 M | 89.89 ± 0.16 | 92.55 ± 0.14 |
| Guo *et al*. [11] | Attention module add-in | Partial VGGNet-16 | 57 M | 91.30± 0.18 | 93.45 ± 0.17 |
| Tong *et al*. [12] | | Single DensnNet-121 | 8.3 M | 92.70± 0.32 | 94.58 ± 0.26 |
| Guo *et al*. [13] | | ResNet-101 | 46.8 M | 89.40 | 91.15 |
| Li *et al*. [15] | | ResNet-18 | 11.7 M | 92.17 ± 0.08 | 92.46 ± 0.09 |
| Tang *et al*. [17] | | Two VGGNet-16 | 276 M | 91.09 ± 0.13 | 92.42 ± 0.16 |
| Li *et al*. [18] | | A ResNet-101 with GRUs | 54.1 M | 92.84 ± 0.36 | 94.26 ± 0.27 |
| Zhang *et al*. [19] | Multiple models | An InceptionV3 with CapsNets | None | 89.03 ± 0.21 | 92.6 ± 0.11 |
| Chen *et al*. [21] | Attention module add-in; Multiple models | A DensnNet-121 with multiple self-designed blocks | None | 93.39 ± 0.39 | 94.95 ± 0.36 |
| Minetto *et al*. [20] | A CNN ensemble | 12 CNNs | None | None | 94.51 ± 0.21 |
| Zhao *et al*. [22] | A CNN ensemble | A DensnNet-121 with four self-designed branches | 23.9 M | 93.05 ± 0.18 | 95.36 ± 0.14 |
| [this work] | Single CNN | EfficeintNet-B0 | 5.3 M | 94.27 ± 0.04 | 95.89 ± 0.08 |

As shown in Table 2, the author's method on NWPU still outperforms, with an outstanding lead on OA but undoubtedly the fewest parameters. Compared to all strategies in detail, the author's lead over the best of feature fusion is 2.6% [3], though the compared method used a smaller testing ratio of 20%; the lead over the best of attention module add-in [12] is 1.3% to 1.6% with a clearly decrease of 36% for parameters; the lead over the best of multiple models [21] is 0.9% to 1.0%, although the compared method's parameter size is not mentioned but undoubtedly be more huge. In addition, the author's lead over the best CNN ensemble [22] is 0.5% to 1.2%, and the improvement on a TR of 10% with more testing examples is more obvious. Based on Eq. (11) and putting the similar comparison results together, we can see that the author's method is more advanced when the testing sets become larger.

Therefore, as a short conclusion, based on all the above OA results on the two benchmark RSI datasets, we can find that the author's method presents a consistent advance compared to the other previous ones; these results also prove that the hypothesis and explanation from mathematical theory, as presented in Section 3.1, are reasonable and persuasive. That is, it is unnecessary to improve the method's computational complexity for transfer learning tasks like RSI-SC.

### 5.3. Confusion Matrixes

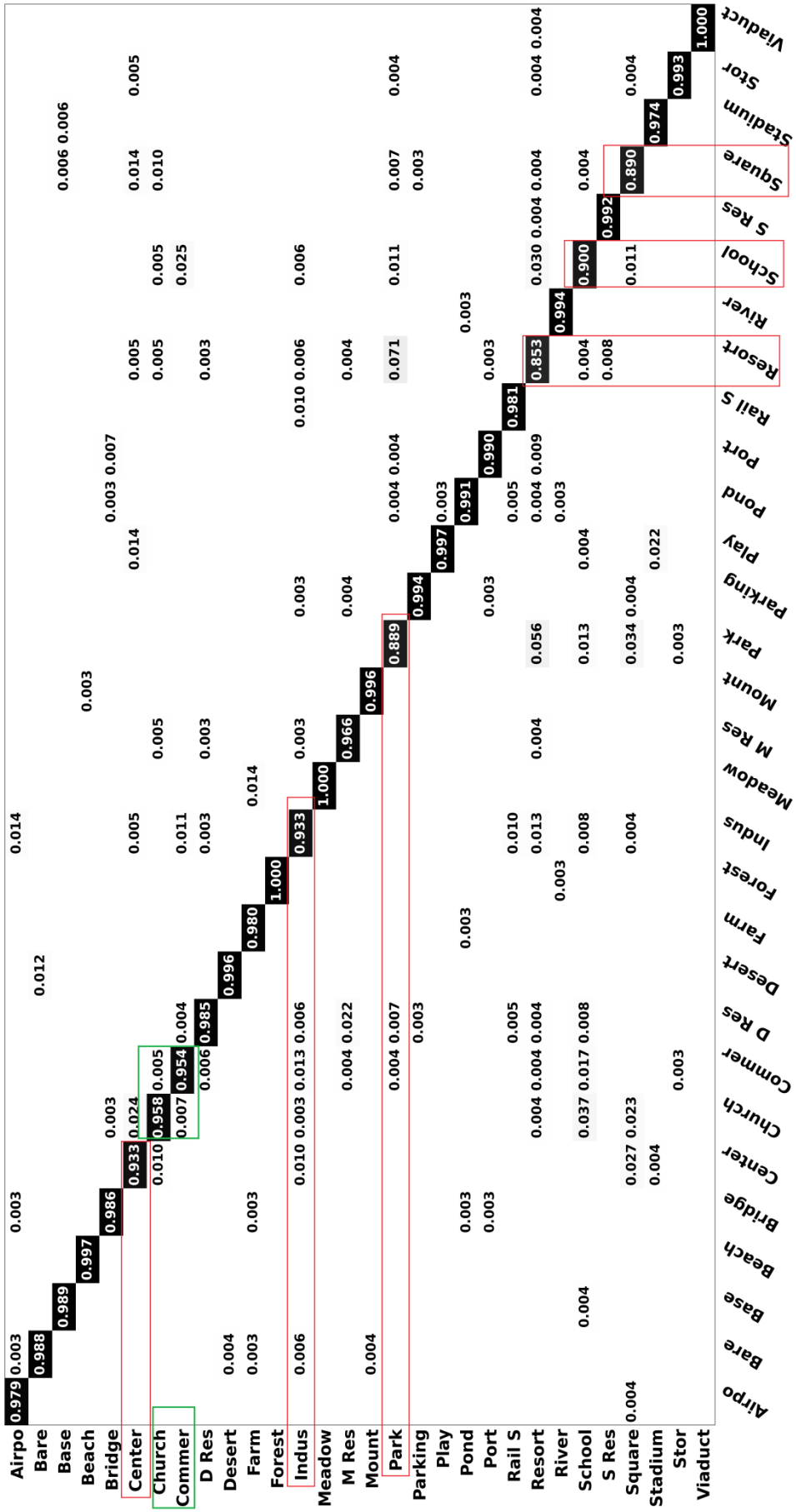



**Figure 8:** Confusion matrix of AID at a TR of 20%.

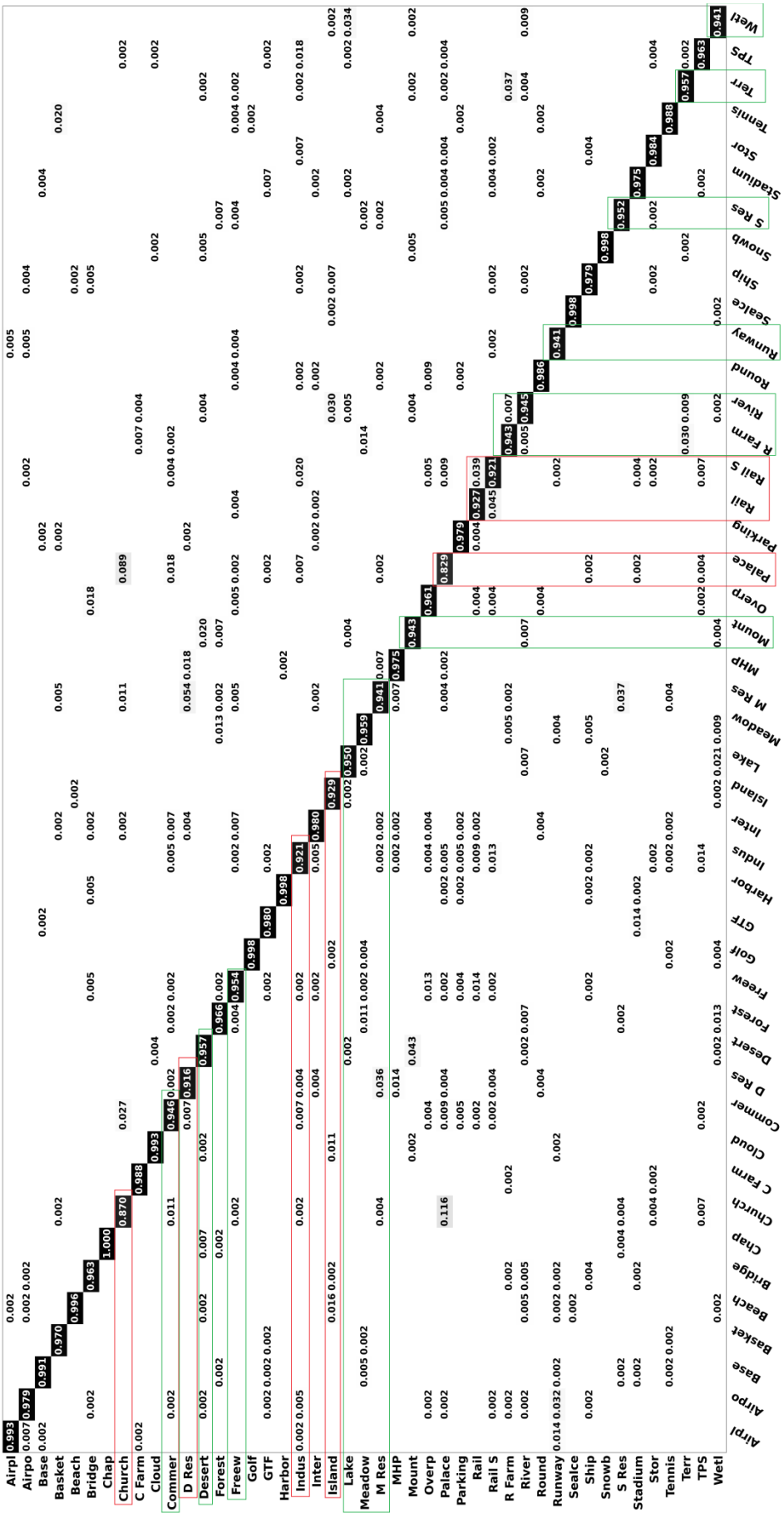



**Figure 9:** Confusion matrix of NWPU at a TR of 20%.

The confusion matrix for AID at a 20% TR is shown in Figure 8, while that for NWPU at a 20% TR is shown in Figure 9. As mentioned before, the author's method has proven to be more advanced with more testing samples. Hence, the matrix of AID at a TR of 20% is shown here in special.

In short, as shown in Figure 8, the model achieved an OA of 97.05% on AID with a TR of 20%, but the confusion results are different among the 30 categories. In detail, the most confusing categories are marked with red rectangles, including center, industry area, park, resort, school, and square, with OA less than 95%; the secondary confusing ones marked with green rectangles have OA slightly less than 97%, including church and commercial area; the other categories' OA are all above 97%. Compared to previous studies [7, 12, 17, 19, 21, 22], we can see that the confusion is consistent, though the author's OA is higher. Giving a quick look to the prior leading methods [21, 22], we can find that these methods have the most confusing subclasses similar to this work, but in particular, compared to the OA results of confusing categories in this work, the OAs in [21] are poorer but those in [22] are higher. In other words, the author's method is more discriminative for all subclasses in AID except the most confusing ones, including park, resort, and square. As mentioned before, a classifier ensemble has the advantage of diversity, but its final performance still depends on whether its individual classifiers are accurate enough. Therefore, the CNN ensemble method in [22] is still suboptimal due to its secondary individual classifiers, though it performs better in the three categories.

In summary, as shown in Figure 9, the model shows an OA of 96.04% on NWPU with a TR of 20%, and still, the confusion results are different among 45 categories. In Figure 9, the most confusing categories marked with red rectangles include church, dense residential, industry area, island, palace, railway, and railway station, with OA less than 94%; the secondary confusing ones are marked with green rectangles with OA less than 96%, including commercial area, desert, freeway, lake, meadow, medium residential, mountain, rectangular farmland, river, runway, sparse residential, terrace, and wetland; the other categories' OA are all above 96%. Looking at the comparable results in [7, 12, 17, 19, 22], we can see that the confusion is still consistent, though the author's OA is higher. Giving the same attention to the priors-leading method in [22], we can find that its most confusing subclasses are still different from this work. That is, compared to the ensemble in [22], the author's method is more discriminative for all subclasses on NWPU except the most confusing ones. However, if compared to the method in [7] with a lower OA of 92.55% for the whole dataset, the confusion results in [22] still show clear OA gaps of approximately 3% to 7% in some categories, including forest, roundabout, tennis court, and so on. Therefore, putting the results on two datasets together, we can see that the poor individual classifiers in the CNN ensemble [22] have made the method suboptimal, even though it has diverse individual classifiers.

In conclusion, taking all the confusion results shown in Figures. 8 and 9 into account, we can see that the author's method surpasses all the other previous methods clearly, with the obvious advantage of cheaper hardware overheads, and as the testing samples increase, the author's method becomes more superior; meanwhile, the most confusing categories are human settlements. Hence, based on the results of the OA and confusion matrixes, we can see that the pre-trained CNN on ImageNet-1K can achieve outstanding performance, though the RSI has clear domain gaps with natural ones.

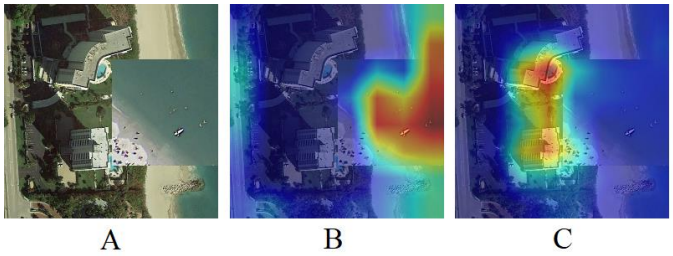
### 5.4. Visualization with Analysis

#### 5.4.1. Class Activation Mapping

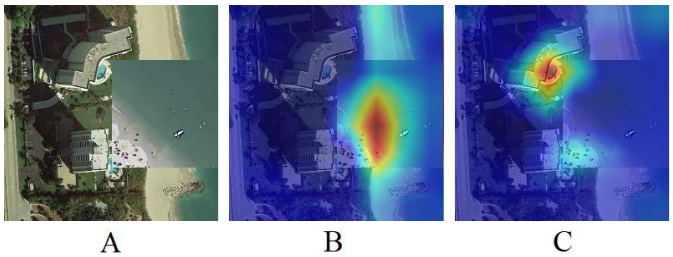
To get a better understanding of CutMix, the author employs activation maps by the GradCAM algorithm [33] to analyze how the CNN's attention is changed for the cut-and-mixed samples, and the maps are shown in Figure 10, in which the A denotes the original scene-mixed images, the B represents activation maps for the beach subclass, and the C represents activation maps for the resort subclass. Note that the brighter areas indicate more discriminative information.

As shown in Figure 10a, the activated area of the EfficientNet-B0 model trained without CutMix is larger and more scattered, which indicates that the model's prediction corresponds to more principal features. On the contrary, as shown in Figure 10b, the activated area of the same model trained with CutMix is smaller and more targeted, and more specifically, as shown in the C part of Figure 10b, the activated area for resort is mainly focused on a swimming pool, which is the most general ground object

of the resort category. Therefore, based on these activation mapping results, we can see that the CutMix has guided the CNN to learn more discriminative features in RSI.




**(a):** Training without CutMix




**(b):** Training with CutMix

**Figure 10:** Activation maps of scene-mixed images derived from Grad-CAM.

### 5.4.2. Stochastic Neighbor Embedding

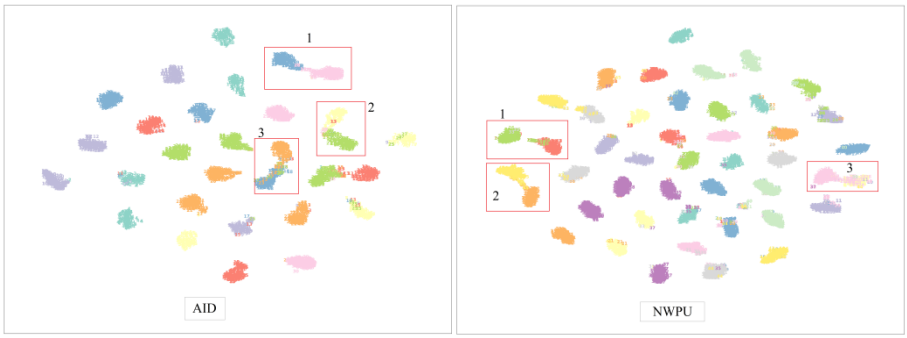



**Figure 11:** t-SNE visualization for AID and NWPU

To get a further verification of the method's effectiveness, this paper employed a technique [34], named t-Distributed Stochastic Neighbor Embedding (t-SNE), to intuitively show the similarity of classified samples, and the visualization results, as shown in Figure 11, have the same category number for AID and NWPU presented in Figures. 4 and 5.

The obviously overlapped category pairs of AID (marked with red rectangles), as shown in the left of Figure 11, include the first pair of playground and stadium, the second pair of center and square, and the third pair of parking and resort. Looking at the right of Figure 11, the clearly overlapped category of NWPU (also marked with red rectangles) contains three pairs, including the first one of desert and mountain, the second one of lake and wetland, and the last one of church and palace. Putting the results for AID and NWPU together, we can see that most of the categories are clearly separated from each other, and the overlapped results are related to the confusion information shown in Figures. 8 and 9. Checking all the previous methods with comparable results [15, 22], we can see that the t-SNE visualization results in this paper are more dispersed both for AID and NWPU, indicating a better classifying result.

## 5.5. Ablation Study

To validate the importance of regularization in training, in the ablation experiments, the whole pipeline of the SC-CNN algorithm is used as the baseline with the OLS and CutMix inactive, and the OA results, as shown in Table 3, include a 20% TR for AID and a 10% TR for NWPU.

The baseline strategy, as shown in row 1 of Table 3, can help the model prevail over all the previous methods in Table 1, with a 0.1% to 6.9% OA increase on AID and NWPU. These results reveal that there has been consistently suboptimal performance in previous studies. The author argues that an

inappropriate training strategy may be the first reason. Looking at the second and third rows of Table 3, we can see that the OLS and CutMix both boost the EfficientNet-B0 performance by a 0.6% OA increase separately. Most importantly, as shown in the last row of Table 3, the combination of OLS and CutMix can boost the model's accuracy by approximately 1.0%. Therefore, all these results prove that regularization is important for RSI-SC, though it has rarely been mentioned in previous studies.

**Table 3.** OA (%) results of ablation studies for regularizations

| DA and regularization | | | AID | NWPU |
|---|---|---|---|---|
| Baseline | OLS | CutMix | TR-20% | TR-10% |
| ✓ | × | × | 95.84 ± 0.20 | 93.36 ± 0.09 |
| ✓ | ✓ | × | 96.44 ± 0.11 | 93.76 ± 0.07 |
| ✓ | × | ✓ | 96.39 ± 0.12 | 93.48 ± 0.07 |
| ✓ | ✓ | ✓ | **96.86** ± 0. 07 | **94.27** ± 0.04 |

**Table 4.** OA (%) results of ablation studies for DA and regularizations

| DA and regularization | | | | | AID | NWPU |
|---|---|---|---|---|---|---|
| Baseline | DA1-2 | DA2-2 | CutMix-1 | CutMix-2 | TR-20% | TR-10% |
| ✓ | ✓ | × | × | ✓ | 96.68 ± 0.11 | 94.13± 0.05 |
| ✓ | ✓ | × | ✓ | × | 96.40 ± 0.10 | 93.96 ± 0.13 |
| ✓ | ✓ | × | ✓ | ✓ | 96.73 ± 0.01 | 94.17 ± 0.12 |
| ✓ | × | ✓ | ✓ | ✓ | 96.81 ± 0.10 | 94.26 ± 0.10 |
| ✓ | × | ✓ | × | ✓ | **96.86** ± 0. 07 | **94.27** ± 0.04 |

To verify the effectiveness of the combination of DAs and regularizations, this work also conducted similar ablation experiments, and the results are shown in Table 4. In detail, as described in Section 3.3, the "DA1" denotes the DA used in Step 1, consisting of the color jitter, horizontal flip, vertical flip, and rotation, and the "DA2" denotes the DA used in Step 2, equal to DA1 but without the color jitter. In addition, the suffixes "-1" or "-2" mean the DAs or regularizations used in Steps 1 or 2, and the baseline is the same as defined in Table 3.

Given the results in rows 1, 2, and 3 of Table 4, we can see that the model's performance degrades both on AID and NWPU, revealing that the training sets transformed by stronger DA have a larger data distribution shift, giving out a suboptimal solution. In particular, the performance degradation is more evident on AID when a stronger DA is active in Step 2 with CutMix inactive, revealing that the impact of intensive DAs on a CNN's performance is greater when the training set is smaller. Comparing the last two rows in Table 4, however, we can see that the model's performance still degrades lightly when CutMix is active in Step 1, meaning that a combination of stronger DAs and regularizations will also result in a suboptimal solution, though more training samples may alleviate the effect. Anyhow, based on the consistent ablation results in Tables 3 and 4, it proves that the proposed combination of DAs and regularizations is effective.

## 6. Discussions

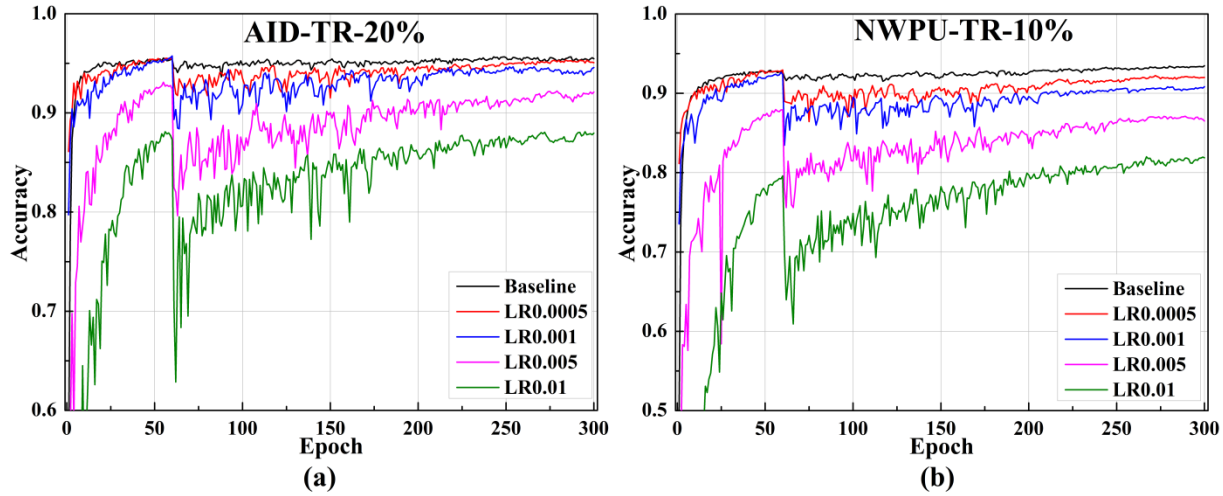



**Figure 12.** Fitting curves for AID and NWPU45D with different LRs

To verify the impact of a larger LR on the CNN's accuracy, this study performs a simple but convincing test. It consists of five different LRs, including 0.0001, 0.0005, 0.001, 0.005, and 0.01, with the same baseline training strategy described in Table 3. The testing results are shown in Figure 11, where the baseline corresponds to a LR of 0.0001.

As shown in Figure 12, it is clear that the model's accuracy declines sharply as the LR grows. We can see that the accuracy drops fast as the LR exceeds 0.001 both for AID and NWPU. The result reveals that the CNN is easier to overfit on a small RSI training set with a larger LR. However, to the author's best knowledge, previous studies in Tables 1 and 2 have not noticed this problem, and some of them have made mistakes.

To verify the impact of adding modules to a pre-trained CNN without re-training on ImageNet-1K again, this study also conducted another simple but persuasive experiment. The test employs a pre-trained EfficientNet-B0 model with all its built-in SE-block parameters re-initialized at random, and then uses the same algorithm presented in Algorithm 1 to train the model both on AID and NWPU. The experiment results, as shown in Table 5, can be directly compared to the related ones in Tables 1 and 2.

Giving a quick look at Table 5, we can see that the same pre-trained CNN will meet significant performance degradation, with OA decreases of 0.15% to 0.24% on AID and those of 0.43% to 0.44% on NWPU, if its pre-trained weights of the SE attention blocks are re-initialized by random; meanwhile, the degradation is more obvious on the larger dataset. These results prove the viewpoints given in Eqs. (5-8), i.e., that pre-training matters a lot if the model's architecture is modified.

**Table 5.** OA (%) results of the attention module pre-training test

| Methods | Architecture | OA | | | |
|---|---|---|---|---|---|
| | | AID-TR20% | AID-TR50% | NWPU-TR10% | NWPU-TR20% |
| SE blocks re-initialized | EfficientNet-B0 | 96.71 ± 0.03 | 97.81 ± 0.05 | 93.84 ± 0.16 | 95.45 ± 0.07 |
| Original pre-trained weights | | 96.86 ± 0. 07 | 98.05 ± 0.06 | 94.27 ± 0.04 | 95.89 ± 0.08 |

Taking the CNN ensemble [22] into account, by adding multiple branches to a pre-trained CNN with self-designed blocks plus built-in attention modules, the authors proposed an ingenious idea to condense the method's complexity and lift the individual classifiers' diversity; however, as proven in Table 5, this architecture modification requires a pre-training on ImageNet-1K again, but it is omitted in fact. Therefore, compared to the author's one, we can see that the method's performance in [22] struggles on the larger NWPU, just like the result in Table 5 behaves.

In general, this work proposed a simple but leading method for classifying RSI by using a lightweight EfficientNet-B0 model. Given the fewer parameters than previous studies, the LS-EfficientNet can perform better in those hardware-restricted fields for classifying RSI, e.g., embedded systems, onboard devices, field tasks, and so on. Given the simple pipeline consisting of an accessible pre-trained CNN model and open source training algorithms, the LS-EfficientNet is also easier to reproduce for those routine tasks for classifying RSI. Based on the following points, however, the LS-EfficientNet still has disadvantages that need improvement. First, putting hardware and time costs aside, the LS-EfficientNet may not be the most cutting-edge method to date. Second, given the experience in this work, a CNN ensemble may have a much better performance than the LS-EfficientNet while still maintaining simplicity and efficiency. Third, the LS-EfficientNet using a pre-trained CNN on ImageNet-1K may achieve suboptimal performance on RSI sets due to the feature's domain gap and other neglected problems. Anyhow, the author will try to propose more efficient methods for RSI-SC in the future.

## 7. Conclusions

In this paper, the author proposes a CNN-based method that aims to provide a leading but efficient solution for RSI-SC by using a lightweight EfficientNet-B0. For this purpose, the paper first investigates several popular strategies in mathematical theory and gives out a qualitative conclusion on these methods' theoretical performance in detail. Based on these findings, the work proposes a novel method using a simple pipeline consisting of a single CNN and its concise training algorithm. Far different from previous studies, the proposed method mainly focuses on tackling the problems, including overfitting, data distribution shift by DA, improper use of training tricks, and other incorrect operations on a pre-trained CNN, which were commonly neglected in previous studies. Compared to the complex and

hardware-extensive ones in previous studies, the proposed method is easy to reproduce due to the fact that all the models, training tricks, and hyperparameter settings are open-sourced. Extensive experiments on two benchmark datasets, including AID and NWPU, show that the proposed method can easily surpass all the previous state-of-the-art ones, with an outstanding accuracy lead of 0.5% to 1.2% if compared to the best prior one in 2022. It should be emphasized that the proposed method has the fewest parameters, which is only 22% of the best competitor in 2022. In addition, ablation test results also prove that the proposed effective combination of training tricks, including OLS and CutMix, can clearly boost a CNN's performance for RSI-SC, with an increase in accuracy of 1.0%.

Taking all the findings in the paper together, the author argues that it is unwise to improve the method's complexity and hardware costs for transfer-leaning tasks like RSI-SC; meanwhile, the consistent suboptimal results in previous studies proven in this paper also make it hard to tell what findings are truly meaningful due to the methods' very close performance.

## References


[1] Fan Hu, Gui-Song Xia, Jingwen Hu and Liangpei Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery", *Remote Sensing*, Print ISSN: 2072-4292, pp. 14680–14707, Vol. 7, No. 11, 5 November 2015, Published by MDPI, DOI: 10.3390/rs71114680, Available: http://www.mdpi.com/2072-4292/7/11/14680.

[2] Souleyman Chaib, Huan Liu, Yanfeng Gu and Hongxun Yao, "Deep feature fusion for vhr remote sensing scene classification", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 4775–4784, Vol. 55, No. 8, 25 May 2017, Published by IEEE, DOI: 10.1109/TGRS.2017.2700322, Available: http://ieeexplore.ieee.org/document/7934005/.

[3] Yishu Liu, Ching Y. Suen, Yingbin Liu and Liwang Ding, "Scene classification using hierarchical wasserstein cnn", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 2494–2509, Vol. 57, No. 5, 28 October 2018, Published by IEEE, DOI: 10.1109/TGRS.2018.2873966, Available: https://ieeexplore.ieee.org/document/8513808/.

[4] Yishu Liu, Yingbin Liu and Liwang Ding, "Scene classification by coupling convolutional neural networks with wasserstein distance", *IEEE Geoscience and Remote Sensing Letters*, Print ISSN: 1545-598X, pp. 722–726, Vol. 16, No. 5, 16 December 2018, Published by IEEE, DOI: 10.1109/LGRS.2018.2883310, Available: https://ieeexplore.ieee.org/document/8579532/.

[5] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo and Junwei Han, "When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 2811–2821, Vol. 56, No. 5, 9 January 2018, Published by IEEE, DOI: 10.1109/TGRS.2017.2783902, Available: http://ieeexplore.ieee.org/document/8252784/.

[6] Yakoub Bazi, Mohamad M. Al Rahhal, Haikel Alhichri and Naif Alajlan, "Simple yet effective fine-tuning of deep cnns using an auxiliary classification loss for remote sensing scene classification", *Remote Sensing*, Print ISSN: 2072-4292, pp. 2908, Vol. 11, No. 24, 5 December 2019, Published by MDPI, DOI: 10.3390/rs11242908, Available: https://www.mdpi.com/2072-4292/11/24/2908.

[7] Jie Xie, Nanjun He, Leyuan Fang and Antonio Plaza, "Scale-free convolutional neural network for remote sensing scene classification", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, 1558-0644, pp. 6916–6928, Vol. 57, No. 9, 27 August 2019, Published by IEEE, DOI: 10.1109/TGRS.2019.2909695, Available: https://ieeexplore.ieee.org/document/8699111/.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai *et al.*, "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale", In *Proceedings of the International Conference on Learning Representations (ICLR)*, 4 May 2021, Vienna, Austria, pp. 1-21, Available: https://openreview.net/forum?id=YicbFdNTTy.

[9] Hugo Touvron, Andrea Vedaldi, Matthijs Douze and Herve Jegou, "Fixing the Train-Test Resolution Discrepancy", *In Proceedings of the Advances in Neural Information Processing Systems* (NeurIPS 2019), 8 December 2019, Vancouver, Canada, ISBN: 978-1-71380-793-3, pp. 8252–8262, Published by Curran Associates Inc., Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/d03a857a23b5285736c4d55e0bb067c8-Paper.pdf.

[10] Mingxing Tan and Quoc Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *In Proceedings of the Proceedings of the 36th International Conference on Machine Learning* (ICML), 9 June 2019, pp. 6105–6114, Published by PMLR, Available: https://proceedings.mlr.press/v97/tan19a.html.

[11] Yiyou Guo, Jinsheng Ji, Xiankai Lu, Hong Huo, Tao Fang *et al.*, "Global-local attention network for aerial scene classification", *IEEE Access*, Print ISSN: 2169-3536, pp. 67200–67212, Vol. 7, 5 June 2019, Published by IEEE, DOI: 10.1109/ACCESS.2019.2918732, Available: https://ieeexplore.ieee.org/document/8721039/.

[12] Wei Tong, Weitao Chen, Wei Han, Xianju Li and Lizhe Wang, "Channel-attention-based densenet network for remote sensing image scene classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Print ISSN: 1939-1404, pp. 4121–4132, Vol. 13, 15 July 2020, Published by IEEE, DOI: 10.1109/JSTARS.2020.3009352, Available: https://ieeexplore.ieee.org/document/9141394/.

[13] Dongen Guo, Ying Xia and Xiaobo Luo, "Scene classification of remote sensing images based on saliency dual attention residual network", *IEEE Access*, Print ISSN: 2169-3536, pp. 6344–6357, Vol. 8, 10 January 2020, Published by IEEE, DOI: 10.1109/ACCESS.2019.2963769, Available: https://ieeexplore.ieee.org/document/8949476/.

[14] Haikel Alhichri, Asma S. Alswayed, Yakoub Bazi, Nassim Ammour and Naif A. Alajlan, "Classification of remote sensing images using efficientnet-b3 cnn model with attention", *IEEE Access*, Print ISSN: 2169-3536, pp. 14078–14094, Vol. 9, 12 January 2021, Published by IEEE, DOI: 10.1109/ACCESS.2021.3051085, Available: https://ieeexplore.ieee.org/document/9320487/.

[15] Jun Li, Daoyu Lin, Yang Wang, Guangluan Xu, Yunyan Zhang *et al.*, "Deep discriminative representation learning with attention map for scene classification", *Remote Sensing*, Print ISSN: 2072-4292, pp. 1366, Vol. 12, No. 9, 26 April 2020, Published by MDPI, DOI: 10.3390/rs12091366, Available: https://www.mdpi.com/2072-4292/12/9/1366.

[16] Hao Sun, Siyuan Li, Xiangtao Zheng and Xiaoqiang Lu, "Remote sensing scene classification by gated bidirectional network", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 82–96, Vol. 58, No. 1, 27 December 2019, Published by IEEE, DOI: 10.1109/TGRS.2019.2931801, Available: https://ieeexplore.ieee.org/document/8844315/.

[17] Xu Tang, Qiushuo Ma, Xiangrong Zhang, Fang Liu, Jingjing Ma *et al.*, "Attention consistent network for remote sensing scene classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Print ISSN: 1939-1404, pp. 2030–2045, Vol. 14, 1 February 2021, Published by IEEE, DOI: 10.1109/JSTARS.2021.3051569, Available: https://ieeexplore.ieee.org/document/9324913/.

[18] Boyang Li, Yulan Guo, Jungang Yang, Longguang Wang, Yingqian Wang *et al.*, "Gated recurrent multiattention network for vhr remote sensing image classification", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 1–13, Vol. 60, 17 January 2022, Published by IEEE, DOI: 10.1109/TGRS.2021.3093914, Available: https://ieeexplore.ieee.org/document/9495118/.

[19] Wei Zhang, Ping Tang and Lijun Zhao, "Remote sensing image scene classification using cnn-capsnet", *Remote Sensing*, Print ISSN: 2072-4292, pp. 494, Vol. 11, No. 5, 28 February 2019, Published by MDPI, DOI: 10.3390/rs11050494, Available: https://www.mdpi.com/2072-4292/11/5/494.

[20] Rodrigo Minetto, Mauricio Pamplona Segundo and Sudeep Sarkar, "Hydra: an ensemble of convolutional neural networks for geospatial land classification", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 6530–6541, Vol. 57, No. 9, 27 August 2019, Published by IEEE, DOI: 10.1109/TGRS.2019.2906883, Available: https://ieeexplore.ieee.org/document/8698456/.

[21] Weitao Chen, Shubing Ouyang, Wei Tong, Xianju Li, Xiongwei Zheng *et al.*, "GCSANet: a global context spatial attention deep learning network for remote sensing scene classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Print ISSN: 1939-1404, pp. 1150–1162, Vol. 15, 11 January 2022, Published by IEEE, DOI: 10.1109/JSTARS.2022.3141826, Available: https://ieeexplore.ieee.org/document/9678028/.

[22] Qi Zhao, Yujing Ma, Shuchang Lyu and Lijiang Chen, "Embedded self-distillation in compact multibranch ensemble network for remote sensing scene classification", *IEEE Transactions on Geoscience and Remote Sensing*, Print ISSN: 0196-2892, pp. 1–15, Vol. 60, 8 November 2022, Published by IEEE, DOI: 10.1109/TGRS.2021.3126770, Available: https://ieeexplore.ieee.org/document/9606819/.

[23] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie *et al.*, "Bag of Tricks for Image Classification with Convolutional Neural Networks", In *Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16 June 2019, Long Beach, California, USA, Available: https://openaccess.thecvf.com/content_CVPR_2019/html/He_Bag_of_Tricks_for_Image_Classification_with_Convolutional_Neural_Networks_CVPR_2019_paper.html.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens and Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision", In *Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), June 2016, Las Vegas, USA, Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html.

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin and David Lopez-Paz, "Mixup: Beyond Empirical Risk Minimization", In *Proceedings of the International Conference on Learning Representations* (ICLR), 3 May 2018, Available: https://openreview.net/forum?id=r1Ddp1-Rb.

[26] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han *et al.*, "Delving deep into label smoothing", *IEEE Transactions on Image Processing*, Print ISSN: 1057-7149, pp. 5984–5996, Vol. 30, 24 June 2021, Published by IEEE, DOI: 10.1109/TIP.2021.3089942, Available: https://ieeexplore.ieee.org/document/9464693/.

[27] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe *et al.*, "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features", In *Proceedings of the IEEE/CVF International*

*Conference on Computer Vision* (ICCV), October 2019, Seoul, Korea, Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Yun_CutMix_Regularization_Strategy_to_Train_Strong_Classifiers_With_Localizable_Features_ICCV_2019_paper.html.

[28] Mingxing Tan and Quoc Le, "EfficientNetV2: Smaller Models and Faster Training", In *Proceedings of the Proceedings of the 38th International Conference on Machine Learning* (ICML), 18 July 2021, pp. 10096–10106, Published by PMLR, Available: https://proceedings.mlr.press/v139/tan21a.html.

[29] Jianming Zhang, Chaoquan Lu, Jin Wang, Xiao-Guang Yue, Se-Jung Lim *et al.*, "Training convolutional neural networks with multi-size images and triplet loss for remote sensing scene classification", *Sensors*, Print ISSN: 1424-8220, pp. 1188, Vol. 20, No. 4, 21 February 2020, Published by MDPI, DOI: 10.3390/s20041188, Available: https://www.mdpi.com/1424-8220/20/4/1188.

[30] Jie Hu, Li Shen and Gang Sun, "Squeeze-and-Excitation Networks", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 18 June 2018, Salt Lake City, USA, pp. 7132–7141, Published by Computer Vision Foundation, Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html.

[31] Gong Cheng, Xingxing Xie, Junwei Han, Lei Guo and Gui-Song Xia, "Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Print ISSN: 1939-1404, pp. 3735–3756, Vol. 13, 20 June 2020, Published by IEEE, DOI: 10.1109/JSTARS.2020.3005403, Available: https://ieeexplore.ieee.org/document/9127795/.

[32] Ilya Loshchilov and Frank Hutter, "Decoupled Weight Decay Regularization", In *Proceedings of the International Conference on Learning Representations* (ICLR), 21 December 2019, New Orleans, Louisiana, United States, Available: https://openreview.net/forum?id=Bkg6RiCqY7.

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh *et al.*, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization", In *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), October 2017, Venice, Italy, pp. 618–626, Published by Computer Vision Foundation, Available: https://openaccess.thecvf.com/content_iccv_2017/html/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.html.

[34] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne", *Journal of Machine Learning Research*, Online ISSN: 1533-7928, pp. 2579–2605, Vol. 9, No. 86, 11 August 2008, Published by Journal of Machine Learning Research. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.