A Robust and Low Complexity Deep Learning Model for Remote Sensing Image Classification

Cam Le[1,3]
cam.levt123@hcmut.edu.vn
HCMC University of Technology
Viet Nam

Lam Pham
Lam.Pham@ait.ac.at
Austrian Institute of Technology
Austria

Nghia NVN
nghianguyenbkdn@gmail.com
Pintel ltd.
South Korea

Truong Nguyen[2,3]
truongnguyen@hcmut.edu.vn
HCMC University of Technology
Viet Nam

Le Hong Trang[1,3]
lhtrang@hcmut.edu.vn
HCMC University of Technology
Viet Nam

ABSTRACT

In this paper, we present a robust and low complexity deep learning model for Remote Sensing Image Classification (RSIC), the task of identifying the scene of a remote sensing image. In particular, we firstly evaluate different low complexity and benchmark deep neural networks: MobileNetV1, MobileNetV2, NASNetMobile, and EfficientNetB0, which present the number of trainable parameters lower than 5 Million (M). After indicating best network architecture, we further improve the network performance by applying attention schemes to multiple feature maps extracted from middle layers of the network. To deal with the issue of increasing the model footprint as using attention schemes, we apply the quantization technique to satisfy the maximum of 20 MB memory occupation. By conducting extensive experiments on the benchmark datasets NWPU-RESISC45, we achieve a robust and low-complexity model, which is very competitive to the state-of-the-art systems and potential for real-life applications on edge devices.

ACM Reference Format:
Cam Le[1,3], Lam Pham, Nghia NVN, Truong Nguyen[2,3], and Le Hong Trang[1,3]. 2022. A Robust and Low Complexity Deep Learning Model for Remote Sensing Image Classification. In *International Conference on Intelligent Information Technology (ICIIT), February 24–26, 2023, Da Nang, Viet Nam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3549555.3549568

1 INTRODUCTION

As the task of remote sensing image classification (RSIC) is considered as an important component in various real-life applications such as urban planning [22, 36], natural hazards detection [27, 39], environmental monitoring [39], vegetation mapping or geospatial object detection [11], it has attracted much research attention in recent years. Indeed, the research community, which focuses on RSIC tasks, has published diverse datasets of remote sensing image as well as proposed a wide range of classification models. The most early dataset of remote sensing image, UCM [52], was publish in 2010. In next years, various remote sensing image datasets were published such as WHU-RS19 [48] in 2012, NWPU VHR-10 [5], SAT6 [1] and RSSCN7 [65] in 2015, SIRI-WHU [59] in 2016, AID [46] and NWPU-RESISC45 [3] in 2017, and OPTIMAL [42] in 2018. Among these datasets, NWPU-RESISC45 [3] presents the largest number of 45 different image scenes and balanced number of 700 images per class. Regarding RSIC systems, they can be separated into two approaches. The first approach mainly focuses on image processing techniques and machine learning based classification. While the image processing techniques are used to extract distinct features from the original image data, the traditional machine learning methods are used to classify these extracted features into certain classes. Regarding image processing based feature extraction, a wide range of methods were proposed such as Texture Descriptors (TD), Color Histogram (CH), Scale-Invariant Feature Transformation (SIFT) [51], wavelet transformation with Gabor/Haar filters [9, 10], bag-of-visual-words (BoVW) based techniques [31, 52]. These methods make effort to transform the original image into a new and condense feature space, likely vector, which is suitable for traditional machine learning classification such as Support Vector Machine (SVM) [9, 52], K-means Clustering [63], or Decision Tree and Neural Network[8]. In the second approach, RSIC research community focuses on deep learning based models, mainly using variants of Deep Convolutional Neural Network (DCNN) such as VGG [53], ResNet [29], DenseNet [38], EfficientNet[55], or Transformer [56]. To train these networks, there are 3 typical strategies[23]: direct training, fine tuning, and using DCNN as a feature extractor. While the first strategy directly trains a network architecture on a RSIC dataset [25], the other two methods make use of pre-trained models on large-scale image datasets to finetune [13, 29, 41] or extract features [17, 18, 20, 25, 58, 61] on a RSIC dataset (i.e. Leveraging a pre-trained models in these two training strategies is considered as the transfer learning technique). As most of datasets of RSIC present a limitation of data compared with natural image datasets such as ImageNet [7], training a network from scratch shows high cost and present ineffective compared with fine tuning methods or using DCNN as a feature extractor.

1. Faculty of Computer Science and Engineering.
2. Faculty of Electrical and Electronics Engineering.
3. HCMC University of Technology (HCMUT) 268 Ly Thuong Kiet, District 10, Ho Chi Minh City, Viet Nam and Vietnam National University Ho Chi Minh City (VNU-HCM) Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam.

ICIIT, February 24–26, 2023, Danang, Viet Nam

ACM ISBN 978-1-4503-9720-9/22/09.
https://doi.org/10.1145/3549555.3549568

Compare between two approaches, the second approach leveraging deep learning based systems proves robust and outperforms the traditional machine learning based approach [19]. However, complicated deep neural networks in the second approach commonly presents very high model complexity which causes challenging for applying RSIC on edge devices. In this paper, we address the problems of those two approaches, aim to develop a robust and low-complexity deep learning model for RSIC task. We mainly contribute:

(1) Firstly, we evaluate and compare current benchmark and low-complexity network architectures: MobileNet, MobileNetV2, NASNetMobile, EfficientB0. Our experimental results indicate that the EfficientNetB0 architecture using the transfer learning technique is more effective for RSIC task.

(2) Secondly, we propose a Multihead attention based layer which is applied to multiple feature maps for improving EfficientNetB0 network performance. To deal with the issue of increasing model complexity using the attention layers, we apply the quantization technique to meet the requirement: The proposed model occupies a maximum of 20 MB which is potential for applying to a wide range of edge devices surveyed in [33].

(3) Finally, we evaluate our best model (EfficientNetB0 network architecture using the transfer learning technique, the proposed Multihead attention based layer, and the quantization technique) on the largest and benchmark dataset of NWPU-RESISC45 [3]. The experimental results show that our proposed RSIC system is competitive to the state of the art, but presents significantly lower model footprint.

2 BACKGROUND

As our proposed deep learning model leverages the parameter-based transfer learning technique and attention schemes, the background of these two techniques is comprehensively presented below.

2.1 The parameter-based transfer learning applied for deep neural network

Humans can be aware that it is easy to transfer knowledge from one domain or task to another. For an instance, it will be easier for a person to learn a second programming language if he/she had experience on a programming language before. In other words, a person can encounter a new task without starting from scratch by leveraging previous experience to learn and adapt to a new task. Inspired by the human capability to transfer knowledge, the machine learning research community has recently focused on the transfer learning techniques and made effort to apply on the computers [44, 64].

In this paper, we apply the parameter-based transfer learning technique, which is very popular and effective for deep neural network network [26]. Given a model of neural network architecture, we firstly define the term of 'pre-trained model': A model was trained on a particular large-scale dataset for a certain task in advance, referred to as the up-stream task. Then, transfer learning is a term that points out the action of applying the pre-trained model for a new task but related in some aspect of the up-stream task. The new task is referred to as the down-stream task. Commonly, the up-stream task is more challenging than the down-stream task (e.g., more objects in tasks of object detection or more categories in classification tasks) and the dataset used in the down-stream task is normally smaller or more specific than the large-scale and general dataset for the up-stream task. The idea and advantages behind the parameter-based transfer learning technique for deep neural network is that utilizing the information gained while solving a challenging up-stream task (i.e. The trainable parameters and the network architecture of the pre-trained model) may not only save time but also enhance the performance on a more simple down-stream task. Regarding the mathematical perspective behind the classification task and deep neural network based model in this paper, it is basically an optimization task which makes gradient descent find the minimum point. Therefore, the starting point of gradient is a very important factor. Indeed, if the starting point of gradient is near the global optimum point, it significantly helps to save the training time as well as avoid the gradient to converse at unexpected local optimization points. By applying the parameter-based transfer learning technique, the distribution of trainable parameters, which is reused from a pre-trained model on an up-stream task, is likely to be near the golden distribution of trainable parameters in a down-stream task rather than random initialization. As the start distribution of trainable parameters is likely same as the golden distribution of trainable parameters, the gradient feasibly converse at very near the global optimal point.

In this paper, we aim to classify remote sensing image into sentiment categories, which is considered as the task of remote sensing image classification (RSIC). As we leverage the parameter-based transfer learning technique, our task of RSIC is referred to as the down-stream task. To solve our down-stream task of RSIC, we there need to define the up-stream task of image classification as well as indicate a pre-trained model with a large-scale dataset. As ImageNet is considered as the benchmark dataset [28] to evaluate a wide range of network architectures on the task of image classification, published pre-trained models on ImageNet from Keras library [6] are considered as the up-stream tasks and leveraged for our down-stream task of RSIC.

2.2 Attention schemes in computer vision

Humans can easily find the important regions in an image. In other words, there are some regions on a image containing specific and distinct features which help humans distinguish from other images. This inspires the computer vision research community focuses on attention mechanisms which help deep learning models know and learn which valuable features. An attention mechanism can be formulated by a function $g(\mathbf{X})$ where $\mathbf{X}$ is the input feature map and $g(\mathbf{X})$ represents a way to create the guidance based on the importance of input feature map $\mathbf{X}$. In other words, the output of $g(\mathbf{X})$ is attention weights which present which region of the input feature map is more important. The attention weights are then element-wise multiplied with the input feature map $\mathbf{X}$ [14, 45] as described by Eq.1

$$f(\mathbf{X}) = g(\mathbf{X}) \odot \mathbf{X} \tag{1}$$

where $f()$ is the attention layer applied on the input feature map $\mathbf{X}$ to generate a new feature map which better presents distinct features, but still retain the original feature map size.

The current attention mechanisms applied to the computer vision research field and deep learning models can be divided into some main groups described in detail below.

Squeeze-and-excitation networks (SE) [14]. It is a channel-based attention mechanism, SE layer focuses on the particular features on the channel dimension. Moreover, SE uses global average pooling (GAP) before feeding to a multi layers perceptron neural network with a sigmoid function at the last layer. Then, it further applies a channel-wise multiplication between the input feature map $\mathbf{X}$ and the output of activation layer. The SE is formulated by Eq. 2

$$f_{SE} = g(mlp(GAP(\mathbf{X}))) \odot \mathbf{X} \tag{2}$$

where $g()$ is sigmoid function, $mlp()$ stands for multi-layers perceptron neural network and GAP is a channel wise global average pooling layer.

Channel attention (CA) layer: CA layer is a variant of SE and it is also a channel-based attention method which has been popularly used in convolutional neural networks [12, 60]. Similar to SE layer, the idea behind the channel attention layer is guiding the model to focus on some particular features on the channel. But it seems to be more powerful as it utilize information from both global max and average pooling layer.

In particular, given three-dimensional input feature map $\mathbf{X} \in R^{W \times H \times C}$ where W, H, and C are width, height, and channel dimensions, the channel attention (CA) applied to the feature map $\mathbf{X}$ can be formulated by:

$$f_{CA} = g(mlp(GAP(\mathbf{X})) + mlp(GMP(\mathbf{X}))) \odot \mathbf{X} \tag{3}$$

where $g()$ is sigmoid function, $mlp()$ is a sharing neural layer (e.g. normally use multi-layers perceptron). GAP and GMP are global average pooling and global max pooling of channel wise, respectively.

Spatial attention (SA) layer: enables the deep neural network focus on distinct features on both width and height dimensions rather than the channel dimension as CA or SE mechanisms. As focusing on the spatial features on width and height dimensions, the channel dimension of a three-dimensional input feature maps $\mathbf{X}$ is firstly reduced by using average pooling and max pooling, create two-dimension feature maps of $\mathbf{X_A}, \mathbf{X_M} \in R^{W \times H}$, respectively. Then, a network layer (e.g., normally a convolutional layer), described by $conv()$ is applied and followed by a Sigmoid function. The SA layer is formulated as Eq. 4

$$f_{SA} = g(conv([\mathbf{X_A}, \mathbf{X_M}])) \odot \mathbf{X} \tag{4}$$

where $g()$ is sigmoid function, $conv()$ represents for a convolutional layer.

Convolutional Block Attention Module (CBAM): While SE/CA and SA mechanisms only focus on either channel features or spatial features, CBAM [45], combines both these attention methods, creates a robust guidance for network to process important regions of a certain feature map. This attention mechanism can be described by formulas: Eq.5 and Eq.6:

$$\mathbf{X'} = f_{CA}(\mathbf{X}) \tag{5}$$

$$f_{CBAM} = f_{SA}(\mathbf{X'}) \tag{6}$$

Multihead self attention (MSA): Unlike above methods which make effort to enhance important regions of a feature map, this attention scheme [40] helps to indicates the similarity score, the dependency between regions in the feature map. In other worlds, Multihead self attention is effective to represent the relation between two regions of a feature map which are closed or far from each others. Regarding the mathematical intuition behind the Multihead self attention, each attention head can be described as mapping a query ($\mathbf{Q}$) and a set of key($\mathbf{K}$)-value($\mathbf{V}$) pairs to an output, where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ obtained through a linear transformation of the input feature map $\mathbf{X}$ as shown in Eq.7, 8, and 9. Then, the output of an attention head can be calculated using Eq. 10.

$$\mathbf{Q} = \mathbf{X} \cdot \mathbf{W_q} \tag{7}$$

$$\mathbf{K} = \mathbf{X} \cdot \mathbf{W_k} \tag{8}$$

$$\mathbf{V} = \mathbf{X} \cdot \mathbf{W_v} \tag{9}$$

$$g_n = softmax(\frac{\mathbf{Q}\mathbf{K}^\mathbf{T}}{\sqrt{\mathbf{d_k}}})\mathbf{V} \tag{10}$$

where g_n is the n^{th} attention head, $\mathbf{W_q}, \mathbf{W_k}, \mathbf{W_v}$ are weight matrices, $\mathbf{K^T}$ is the transpose of $\mathbf{K}$ and $\mathbf{d_k}$ is the number of key dimension which is one of the dimension of the weight matrices.

As each attention head learns a different set of weight matrices, they will be different from each others. Therefore, when joining many self attention heads together followed by a linear transformation or an addition operation as an ensemble of multiple heads, it forms a Multihead self attention layer which helps to learn an input feature map better. A Multihead self attention layer with N heads which is applied on the input feature map $\mathbf{X}$ is described by

$$f_{MA} = \sum_{n=1}^{N} g_n \odot \mathbf{X} \tag{11}$$

3 PROPOSED DEEP LEARNING BASED SYSTEM FOR RSIC TASK

Overall, the high-level architecture of our proposed deep learning based system for RSIC task is presented in Figure 1. As Figure 1 shows, the proposed RSIC system is separated into two main parts: data augmentation methods and a deep neural network for classification.

3.1 Data augmentation methods

In this paper, we apply five data augmentation methods: Image Rotation (IR) [30], Random Cropping (RC) [30], Random Erasing (RE) [34], Random Noise Addition (RNA), and Mixup (Mi) [37, 50] to the remote sensing image input data. As Random Cropping (RC) [30], Random Erasing (RE) [34], Random Noise Addition (RNA), and Mixup (Mi) [37, 50] are used on batches of images during the training process, they are referred to as the online data augmentation methods. Meanwhile, Image Rotation (IR) [30] is referred to as the offline data augmentation as this method is applied on the original dataset before the training process.

Figure 1: The high level architecture of proposed RSIC system

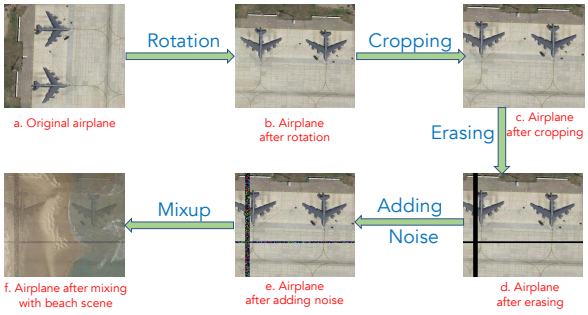

Figure 2: Data augmentation methods: Rotation, Random Cropping, Random Erasing, Adding Noise, and Mixup in the order.

Initially, all images in the original dataset are rotated using three different angles: 90, 180, and 270, respectively. This data augmentation method is referred to as Image Rotation (IR) and an example of IR method with an angle of 90 degree is shown in Figure 2 (b). As three angles mentioned are used, we obtain a new dataset which is four times larger than the original dataset (i.e. the original images and three new images generated by Image Rotation method with three angles). Next, batches of 60 images are randomly selected from the new dataset. For each batch, we apply Random Cropping (RC) [30], Random Erasing (RE) [34], Random Noise Addition (RNA), and Mixup (Mi) [37, 50] methods, respectively. Firstly, images in a batch are randomly cropped with a reduction of 10 pixels on both of width and height dimensions as shown in Figure 2 (c) (i.e., The channel dimension is retained), referred to as Random Cropping (RC). Next, on both width and height dimensions of each image, 20 random and continuous pixels are erased as shown in Figure 2 (d), referred to as Random Erasing (RE). The cropped and erased images are then added by a random noise which is generated from Gaussian distribution as shown in Figure 2 (e), referred to as Random Noise Addition (RNA). Finally, the images are mixed together with random ratios as shown in Figure 2 (f), referred to as Mixup (Mi). As both Uniform and Beta distributions are used to generate the mixup ratios as well as we use both the original image and the new mixup images, the batch size increases three times from 60 to 180 images.

3.2 Apply the transfer learning technique for our proposed deep neural network classification

As Figure 1 shows, our proposed deep learning model for classification can be separated into two main parts: The convolutional neural network based backbone (CNN based backbone) and the multilayer perceptron based classification (MLP-based classification). While the CNN based backbone helps to transfer the input images to condensed feature maps, the MLP based classification classifies these condensed feature maps into certain categories.

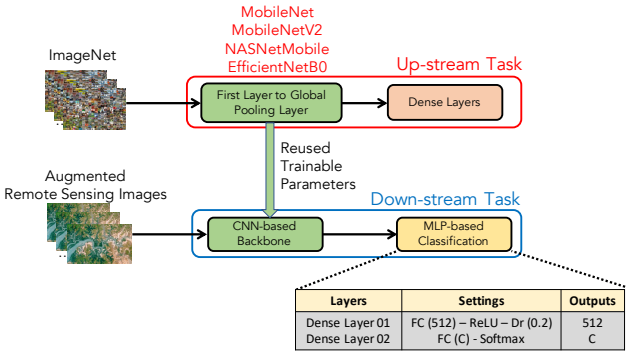

Figure 3: Apply the transfer learning technique for the proposed deep neural network classification

To indicate which CNN based backbone is effective for RSIC task, we evaluate different benchmark deep neural network architectures which are available in Keras library [6]. As we aim to achieve a low-complexity model for RSIC which is lower than 5 M of trainable parameters, only four network architectures of MobileNetV1, MobileNetV2, NASNetMobile, and EfficientNetB0 from Keras library [6] are evaluated. To leverage these network architectures, we apply the parameter-based transfer learning technique which is mentioned in Section 2.1. The transfer learning process is mainly described in Figure 3. In particular, the benchmark networks of MobileNetV1, MobileNetV2, NASNetMobile, and EfficientNetB0 as described in the higher part of Figure 3 are firstly trained with the large scale dataset of ImageNet, referred to as the up-stream task. Next, only the first layer to the global pooling layer of these pre-trained models are re-used and considered as the CNN-based backbone. The CNN based backbone is then connected with a MLP-based classification, create an end-to-end neural network model for the down-stream task on the target RSIC dataset.

The MLP-based classification as shown in the bottom-right part in Figure 3 performs two dense layers (Dense Layer 01 and 02). The first dense layer comprises one fully connected layer (FC(channel number=512)) followed by a rectified linear unit (ReLU) [21] and a dropout (Dr(drop ratio)) [32]. Meanwhile, the second dense layer uses Softmax layer after the fully connected layer. Notably, the number of channels at the second fully connected layer is set to C that presents the number of categories in the target RSIC dataset.

3.3 Apply attention schemes and explore multiple feature maps to further improve the proposed RSIC system

To further improve the proposed RSIC system, we apply different attention schemes mentioned in Section 2.2 to feature maps extracted from middle layers of the CNN based backbone as shown in Figure 4. The feature maps are the final outputs of convolutional blocks of the CNN based backbone. For an example, EfficientB0

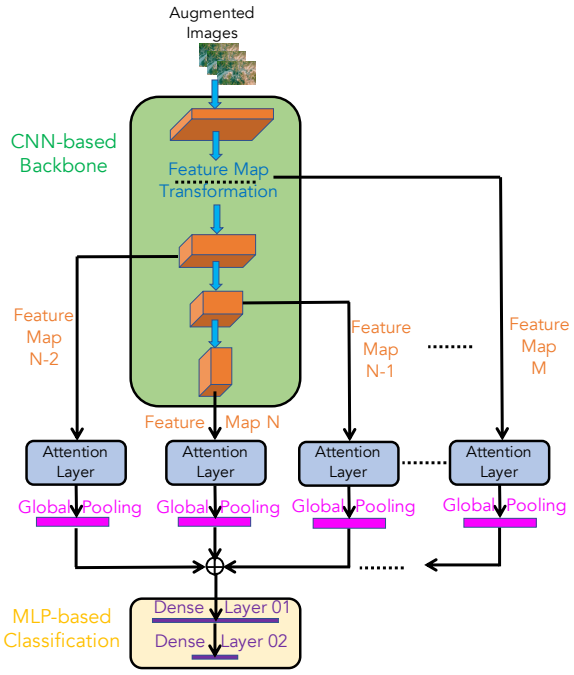

Figure 4: Apply attentions schemes to further improve the proposed deep neural network classification

based backbone presents 7 convolutional blocks, namely block 1 to block 7 [35]. Regarding the attention layer used in Figure 4, we evaluate three types of attention schemes: SE, CBAM, and Multihead attention. The first two attention layers of SE and CBAM are constructed basing on the formulations mentioned Section 2.2. For the Multihead attention scheme, we propose a Multihead attention based layer as shown in Figure 5 which addresses drawbacks of SE or SA (i.e. SE and SA focus on either channel feature or spatial feature). In particular, given an input feature map $\mathbf{X}$ with a size of [W×H×C] where W, H, and C presents width, height, and channel dimensions, we reduce the size of feature map $\mathbf{X}$ across three dimensions using both max and average pooling layers. We then generate 3 matrices: [W× H], [H× C], [W×C]. Next we feed all generated matrices into thee Multihead attention to obtain attention score matrices. Then, we reshape the attention score matrices into the sizes of [W×H×1], [1×H×C],[W×1×C] respectively and element-wise multiply each of them with the input feature map $\mathbf{X}$. Finally, we conduct an average of three results of multiplications, generate the output tensor $\mathbf{Y}$ with the size of [W×H×C] which is same size as the input feature map $\mathbf{X}$. Notably, we set the number of heads to 32 and set the key dimension to 8 for each Multihead attention. By applying our proposed Multihead attention based layer, both channel feature (feature maps with sizes of [H× C], [W×C]) and spatial feature (feature map with size of [W× H]) are focused, which help the network learn distinct features from the input feature map better.

As SE, CBAM, or our proposed Multihead attention base layers only transforms an input feature map $\mathbf{X}$ to an output feature map $\mathbf{Y}$ and retains the size of the input feature maps, we then apply a

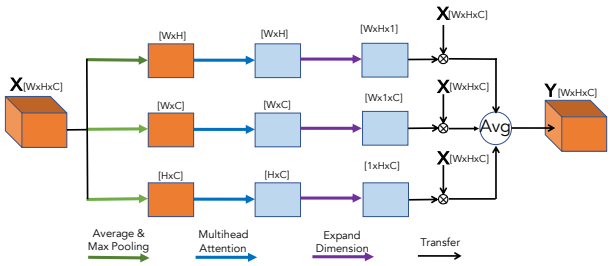

Figure 5: Proposed Multihead attention based layer.

global average pooling layer after each attention layer to scale down the feature maps to vectors which is suitable for the MLP-based classifier for classification as shown in Figure 3.

4 EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Dataset

In this paper, we evaluate the benchmark dataset NWPU-RESISC45[4]. NWPU-RESISC45 dataset was collected from more than 100 countries and regions in the world, consists of 31,500 remote sensing images. The remote sensing images are separated into 45 scene classes, each of which 700 images in RGB color format with resolution of $256 \times 256 \times 3$ To compare with the state-of-the-art systems, we obey the original settings mentioned in [4]. We then split the NWPU-RESISC45 dataset into Training and Testing subsets with two different ratios: 20%-80% and 10%-90%, respectively.

4.2 Evaluation Metrics

To compare with the state-of-the-art systems, Accuracy (Acc.%) is used as the main metric, which was proposed in almost benchmark datasets of AID[47], UCM[52], or NWPU-RESISC45[4].

4.3 Model implementation and settings

As the data augmentation method of Mixup [37] is applied, the ground truth are not in one-hot encoding format. We therefore apply Kullback-Leibler divergence (KL) loss [16] instead of Entropy loss.

$$Loss_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left\{\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right\} + \frac{\lambda}{2}||\Theta||_2^2, \tag{12}$$

where Θ presents trainable parameters, the constant λ is empirically set to 0.0001, the batch size N is set to 60, $\mathbf{y_i}$ and $\hat{\mathbf{y}}_\mathbf{i}$ denote expected and predicted results, respectively. We construct proposed deep learning networks with Tensorflow framework using Adam [15] for optimization. The training and evaluating processes are conducted on two GPU Titan RTX 24GB. The training process is stopped after 60 epoches. While the first 50 epoches uses the learning rate of 0.0001 and all data augmentation methods mentioned in Section 3.1, the remaining 10 epoches uses the lower learning rate of 0.000001 with only the offline Random Rotation data augmentation method

4.4 Experimental results

According to the results shown in Table 2, the proposed RSIC system using the transfer learning technique and EfficientNetB0 and

Table 1: Performance (Acc.%) of EfficientNetB0 with the proposed Multihead attention applied for feature maps extracted from different convolutional blocks on the benchmark NWPU-RESISC45 dataset using 20%-80% splitting settings.

Convolutinal Blocks	Accuracy (%)	Parameters (M)
Block 7	93.1	6.0
Blocks 6 to 7	93.0	7.5
Blocks 5 to 7	93.8	9.4
Blocks 4 to 7	92.8	11.2
Blocks 3 to 7	92.5	13.3

Table 2: Performance comparison among benchmark network architectures, with the transfer learning technique and without attention scheme, on the benchmark NWPU-RESISC45 dataset using 20%-80% splitting settings.

Network	Accuracy (%)	Parameters (M)
MobileNet	90.2	3.7
MobileNetV2	90.9	2.9
NASNetMobile	91.7	4.8
EfficientNetB0	92.0	4.6

NASNetMobile based architectures are competitive and outperform MobileNet and MobileNetV2. As EfficientNetB0 accuracy (92.0%) is not only better than NASNetMobile (91.7%) but EfficientNetB0 footprint (4.6 M) is also smaller than NASNetMobile (4.8 M), we select EfficientNetB0 architecture for further experiments.

Given EfficientNetB0 backbone, we evaluate our proposed RSIC system applying three types of attention layers: SE, CBAM, and the proposed Multihead attention. In this experiment, only thee feature maps which are extracted from the final three convolutional bocks (block 5 to block 7) in the EfficientNetB0 backbone are used. As Table 3 shows, applying attention layers helps to improve the system performance by 0.1%, 0.3%, and 1.8% with SE, CBAM, and the proposed Multihead attention, respectively.

As the proposed Multihead attention layer outperforms SE and CBAM layers, we then evaluate the proposed Multihead attention with different number of feature maps. As the results are shown in Table 1, using three feature maps still achieves the best performance. Regarding the model complexity, using the proposed Multihead attention layer with three feature maps increases the model footprint from 4.6 M to 9.4 M parameters. To meet the constraints of maximum 20 MB of memory occupation, we apply the quantization technique which helps to reduce the model complexity to 9.4 MB (i.e. The quantization technique help to quantize a 32-bit floating point to 8-bit integer, then reduce the model footprint to 1/4 of the original footprint). Notably, although the pruning techniques can help to significantly reduce a deep learning model to 1/10 of the original size [24], pruning parameters considered as zero still occupy the memorize of edge devices and cost the same computation as the non-pruning parameters. Therefore, the pruning technique is not applied in this paper.

By using EfficientNetB0 as CNN-based backbone, the transfer learning, the proposed Multihead attention layer for three feature maps and the quantization technique, we achieve a low-complexity RISC model (9.4 MB). We evaluate this model on NWPU-RESISC45 with two splitting settings as mentioned in Section 4.1 and compare with the state-of-the-art systems. As Table 4 shows, we can see that our results are very competitive compared with the state-of-the-art systems. We achieve accuracy scores of 91.0% and 93.8% with training proportions of 10% and 20% respectively. Compared with the system also using EfficientNetB0 in [2], our proposed RSIC not only outperforms but also presents a lower model footprint. Our proposed system performs lower than 2% compared with the best model using a Transformer based architecture [57]. However, our model presents a significantly low memory occupation (9.4 M/9.4 MB) compared with the Transformer based model.

Table 3: Performance comparison of EfficientB0 with the transfer learning and different attention schemes on the benchmark NWPU-RESISC45 dataset using 20%-80% splitting settings.

Attention	SE	CBAM	Proposed Multihead
Accuracy (%)	92.1	92.3	93.8
Parameters (M)	10.4	6.7	9.4

Table 4: Performance (Acc.%) comparison to the state-of-the-art systems on the benchmark NWPU-RESISC45 dataset with two splitting settings.

Methods	10% training	20% training
MG-CAP (Log-E) (55.99 M) [43]	89.4	91.7
MG-CAP (Bilinear) (55.99 M) [43]	89.4	93.0
MG-CAP (Sqrt-E) (55.99 M) [43]	90.8	93.0
EfficientNet-B0-aux ($\approx 5.3M$) [2]	90.0	92.9
EfficientNet-B3-aux ($\approx 13M$) [2]	91.1	93.8
VGG-16 + MTL ($\approx$ 138.4 M) [62]	-	91.5
ResNeXt-50 + MTL ($\approx$ 25 M) [62]	-	93.8
ResNeXt-101 + MTL ($\approx$ 88.79 M) [62]	91.9	94.2
SE-MDPMNet (5.17 M) [54]	91.8	94.1
LGRIN (4.63 M) [49]	91.9	94.4
Transformer (46.3 M) [57]	93.1	95.6
Our systems (9.4 M / 9.4 MB)	**91.0**	**93.8**

5 CONCLUSION

This paper has presented a deep learning based model for remote sensing image classification (RSIC). By conducting extensive experiments, we indicate that applying multiple techniques of transfer learning, Multihead attention on multiple feature maps, and quantization to EfficientNetB0 based network architecture helps to achieve a high-performance and low-complexity RSIC system. The experimental results prove our proposed RSIC system competitive to the state-of-the-art systems and potential to apply on a wide range of edge devices.

ACKNOWLEDGMENTS

We would like to thank Ho Chi Minh City University of Technology (HCMUT), Vietnam National University Ho Chi Minh City (VNU-HCM) for the support of time and facilities for this study.

REFERENCES

[1] Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. 2015. Deepsat: a learning framework for satellite imagery. In *Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems*. 1–10.

[2] Yakoub Bazi, Mohamad M Al Rahhal, Haikel Alhichri, and Naif Alajlan. 2019. Simple yet effective fine-tuning of deep CNNs using an auxiliary classification loss for remote sensing scene classification. *Remote Sensing* 11, 24 (2019), 2908.

[3] Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* 105, 10 (2017), 1865–1883.

[4] Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* 105, 10 (2017), 1865–1883.

[5] Gong Cheng, Junwei Han, Peicheng Zhou, and Lei Guo. 2014. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing* 98 (2014), 119–132.

[6] François Chollet et al. 2015. Keras. https://keras.io.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

[8] Peijun Du, Junshi Xia, Wei Zhang, Kun Tan, Yi Liu, and Sicong Liu. 2012. Multiple classifier system for remote sensing image classification: A review. *Sensors* 12, 4 (2012), 4764–4792.

[9] Hela Elmannai, MohamedAnis Loghmari, and Mohamed Saber Naceur. 2016. A new classification approach based on source separation and feature extraction. In *2016 International Symposium on Signal, Image, Video and Communications (ISIVC)*. IEEE, 137–141.

[10] Hela Elmannai, Mohamed Anis Loghmari, and Mohamed Saber Naceur. 2013. Support vector machine for remote sensing image classification. In *Proceedings Engineering & Technology*, Vol. 2. 68–72.

[11] Quanlong Feng, Jiantao Liu, and Jianhua Gong. 2015. UAV remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote sensing* 7, 1 (2015), 1074–1094.

[12] Yiyou Guo, Jinsheng Ji, Xiankai Lu, Hong Huo, Tao Fang, and Deren Li. 2019. Global-local attention network for aerial scene classification. *IEEE Access* 7 (2019), 67200–67212.

[13] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing* 7, 11 (2015), 14680–14707.

[14] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. *CoRR* abs/1412.6980 (2015).

[16] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.

[17] Boyang Li, Yulan Guo, Jungang Yang, Longguang Wang, Yingqian Wang, and Wei An. 2021. Gated recurrent multiattention network for VHR remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–13.

[18] Fengpeng Li, Ruyi Feng, Wei Han, and Lizhe Wang. 2020. An augmentation attention mechanism for high-spatial-resolution remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), 3862–3878.

[19] Maryam Mehmood, Ahsan Shahzad, Bushra Zafar, Amsa Shabbir, and Nouman Ali. 2022. Remote sensing image classification: A comprehensive review and applications. *Mathematical Problems in Engineering* 2022 (2022).

[20] Rodrigo Minetto, Maurício Pamplona Segundo, and Sudeep Sarkar. 2019. Hydra: An ensemble of convolutional neural networks for geospatial land classification. *IEEE Transactions on Geoscience and Remote Sensing* 57, 9 (2019), 6530–6541.

[21] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*.

[22] Maik Netzband, William L Stefanov, and Charles Redman. 2007. *Applied remote sensing for urban planning, governance and sustainability*. Springer Science & Business Media.

[23] Keiller Nogueira, Otávio AB Penatti, and Jefersson A Dos Santos. 2017. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition* 61 (2017), 539–556.

[24] Nicolas Pajusco, Richard Huang, and Nicolas Farrugia. 2020. Lightweight Convolutional Neural Networks on Binaural Waveforms for Low Complexity Acoustic Scene Classification.. In *DCASE*. 135–139.

[25] Lam Pham, Khoa Tran, Dat Ngo, Jasmin Lampert, and Alexander Schindler. 2022. Remote Sensing Image Classification using Transfer Learning and Attention Based Deep Neural Network. *arXiv preprint arXiv:2206.13392* (2022).

[26] Rafael Pires de Lima and Kurt Marfurt. 2019. Convolutional neural network for remote-sensing scene classification: Transfer learning analysis. *Remote Sensing* 12, 1 (2019), 86.

[27] Dimitris Poursanidis and Nektarios Chrysoulakis. 2017. Remote Sensing, natural hazards and the contribution of ESA Sentinels missions. *Remote Sensing Applications: Society and Environment* 6 (2017), 25–38.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 3 (2015), 211–252.

[29] Amsa Shabbir, Nouman Ali, Jameel Ahmed, Bushra Zafar, Aqsa Rasheed, Muhammad Sajid, Afzal Ahmed, and Saadat Hanif Dar. 2021. Satellite and scene image classification based on transfer learning and fine tuning of ResNet50. *Mathematical Problems in Engineering* 2021 (2021).

[30] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.

[31] Harini Sridharan and Anil Cheriyadat. 2014. Bag of lines (BoL) for improved aerial scene representation. *IEEE Geoscience and Remote Sensing Letters* 12, 3 (2014), 676–680.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[33] Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. 2021. Mind your weight (s): A large-scale study on insufficient machine learning model protection in mobile apps. In *30th USENIX Security Symposium (USENIX Security 21)*. 1955–1972.

[34] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. 2020. Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 9 (2020), 2917–2931.

[35] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[36] Rajesh Bahadur Thapa and Yuji Murayama. 2009. Urban mapping, accuracy, & image classification: A comparison of multiple approaches in Tsukuba City, Japan. *Applied geography* 29, 1 (2009), 135–144.

[37] Y. Tokozume, Y. Ushiku, and T. Harada. 2018. Learning from between-class examples for deep sound recognition. *in ICLR* (2018).

[38] Wei Tong, Weitao Chen, Wei Han, Xianju Li, and Lizhe Wang. 2020. Channel-attention-based DenseNet network for remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), 4121–4132.

[39] Cees J Van Westen. 2013. Remote sensing and GIS for natural hazards assessment and disaster risk management. *Treatise on geomorphology* 3 (2013), 259–298.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[41] Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. 2022. An Empirical Study of Remote Sensing Pretraining. *IEEE Transactions on Geoscience and Remote Sensing* (2022).

[42] Qi Wang, Shaoteng Liu, Jocelyn Chanussot, and Xuelong Li. 2018. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 57, 2 (2018), 1155–1167.

[43] Shidong Wang, Yu Guan, and Ling Shao. 2020. Multi-granularity canonical appearance pooling for remote sensing scene classification. *IEEE Transactions on Image Processing* 29 (2020), 5396–5407.

[44] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. *Journal of Big data* 3, 1 (2016), 1–40.

[45] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[46] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. 2017. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing* 55, 7 (2017), 3965–3981.

[47] Gui-Song Xia, Jingwen Hu, Fan Hu, Baoguang Shi, Xiang Bai, Yanfei Zhong, Liangpei Zhang, and Xiaoqiang Lu. 2017. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing* 55, 7 (2017), 3965–3981.

[48] Gui-Song Xia, Wen Yang, Julie Delon, Yann Gousseau, Hong Sun, and Henri MaÎtre. 2010. Structural high-resolution satellite image indexing. *Symposium: 100 Years ISPRS - Advancing Remote Sensing Science*.

[49] Chengjun Xu, Guobin Zhu, and Jingqian Shu. 2021. A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–15.

[50] Kele Xu, Dawei Feng, Haibo Mi, Boqing Zhu, Dezhi Wang, Lilun Zhang, Hengxing Cai, and Shuwen Liu. 2018. Mixup-based acoustic scene classification using multi-channel convolutional neural network. In *Pacific Rim Conference on Multimedia*. 14–23.

[51] Yi Yang and Shawn Newsam. 2008. Comparing SIFT descriptors and Gabor texture features for classification of remote sensed imagery. In *2008 15th IEEE international conference on image processing*. IEEE, 1852–1855.

[52] Yi Yang and Shawn Newsam. 2010. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 270–279.

[53] Mu Ye, Ni Ruiwen, Zhang Chang, Gong He, Hu Tianli, Li Shijun, Sun Yu, Zhang Tong, and Guo Ying. 2021. A Lightweight Model of VGG-16 for Remote Sensing Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), 6916–6922.

[54] Bin Zhang, Yongjun Zhang, and Shugen Wang. 2019. A lightweight and discriminative model for remote sensing scene classification with multidilation pooling module. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 12, 8 (2019), 2636–2653.

[55] Deyuan Zhang, Zhenghong Liu, and Xiangbin Shi. 2020. Transfer learning on EfficientNet for remote sensing image classification. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. IEEE, 2255–2258.

[56] Jianrong Zhang, Hongwei Zhao, and Jiao Li. 2021. TRS: Transformers for remote sensing scene classification. *Remote Sensing* 13, 20 (2021), 4143.

[57] Jianrong Zhang, Hongwei Zhao, and Jiao Li. 2021. TRS: Transformers for Remote Sensing Scene Classification. *Remote Sensing* 13, 20 (2021), 4143.

[58] Xinqi Zhang, Weining An, Jinggong Sun, Hang Wu, Wenchang Zhang, and Yaohua Du. 2021. Best representation branch model for remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), 9768–9780.

[59] Bei Zhao, Yanfei Zhong, Gui-Song Xia, and Liangpei Zhang. 2015. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing* 54, 4 (2015), 2108–2123.

[60] Qi Zhao, Yujing Ma, Shuchang Lyu, and Lijiang Chen. 2022. Embedded Self-Distillation in Compact Multibranch Ensemble Network for Remote Sensing Scene Classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–15. https://doi.org/10.1109/tgrs.2021.3126770

[61] Zhicheng Zhao, Jiaqi Li, Ze Luo, Jian Li, and Can Chen. 2020. Remote sensing image scene classification based on an enhanced attention module. *IEEE Geoscience and Remote Sensing Letters* 18, 11 (2020), 1926–1930.

[62] Zhicheng Zhao, Ze Luo, Jian Li, Can Chen, and Yingchao Piao. 2020. When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework. *Remote Sensing* 12, 20 (2020), 3276.

[63] Jian Zheng, Zhanzhong Cui, Anfei Liu, and Yu Jia. 2008. A K-means remote sensing image classification method based on AdaBoost. In *2008 Fourth International Conference on Natural Computation*, Vol. 4. IEEE, 27–32.

[64] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.

[65] Qin Zou, Lihao Ni, Tong Zhang, and Qian Wang. 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters* 12, 11 (2015), 2321–2325.