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

Cam Le1,3
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 Nguyen2,3
truongnguyen@hcmut.edu.vn
HCMC University of Technology
Viet Nam

Le Hong Trang1,3
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 a number of trainable parameters lower than 5 Million (M) or occupy 20 MB memory. After indicating the 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 due to using attention schemes, we apply the quantization technique to satisfy the maximum memory occupation of 20 MB. 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.

CCS CONCEPTS
• Computing methodologies → Artificial intelligence.

KEYWORDS
Deep learning, convolutional neural network (CNN), remote sensing image classification (RSIC), data augmentation, model complexity.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICICIT 2023, February 24–26, 2023, Da Nang, Vietnam
© 2023 Copyright held by the owner(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9961-6/23/02...
https://doi.org/10.1145/3591569.3591601

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 images as well as proposed a wide range of classification models. The most early dataset of remote sensing images, UCM [52], was published in 2010. In the 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 a 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 image processing techniques are used to extract distinct features from the original image data, 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 condensed 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...
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 instance, it will be easier for a person to learn a second programming language if he/she had experience in 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 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 [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 to 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 descent is a very important factor. Indeed, if the starting point of the gradient is near the global optimum point, it significantly helps to save the training time as well as avoid the gradient to converge 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 starting distribution of trainable parameters is likely the same as the golden distribution of trainable parameters, the gradient is feasible to converge at very near the global optimal point.

In this paper, we aim to classify remote sensing images 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 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 an image containing specific and distinct features which help humans distinguish from other images.
This inspires the computer vision research community to focus on attention mechanisms which help deep learning models know and learn which valuable features. An attention mechanism can be formulated by a function \( g(X) \) where \( X \) is the input feature map and \( g(X) \) represents a way to create the guidance based on the importance of input feature map \( X \). In other words, the output of \( g(X) \) 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 \( X \) as described by Eq. 1:

\[
f(X) = g(X) \odot X
\]

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

The current attention mechanisms applied to 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, which focuses on the particular features on the channel dimension. Overall, the SE method is formulated by Eq. 2:

\[
f_{SE} = g(mlp(GAP(X))) \odot X
\]

where \( g() \) is sigmoid function, \( mlp() \) stands for multi-layers perceptron neural network and \( GAP \) is a global average pooling. Initially, a global average pooling \( GAP \) is applied to the input feature map before feeding into a multi layers perceptron neural network with a sigmoid function at the last layer. Then, a channel-wise multiplication between the input feature map \( X \) and the output of the sigmoid activation layer is applied.

**Channel attention (CA):** CA mechanism 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, the idea behind the CA is guiding the model to focus on some particular features on the channel, but CA utilizes information from both global max and average pooling layers. In particular, given three-dimensional input feature map \( X \in \mathbb{R}^{W \times H \times C} \) where \( W, H, \) and \( C \) are width, height, and channel dimensions, the channel attention (CA) applied to the feature map \( X \) can be formulated by:

\[
f_{CA} = g(mlp(GAP(X) + mlp(GMP(X)))) \odot X
\]

where \( g() \) is a sigmoid function, \( mlp() \) is a channelwise global average pooling layer. Initially, a global average pooling \( GAP \) is applied to the input feature map before feeding into a multi layers perceptron neural network with a sigmoid function at the last layer. Then, a channel-wise multiplication between the input feature map \( X \) and the output of the sigmoid activation layer is applied.

**Spatial attention (SA):** enables the deep neural network to 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 map \( X \) is firstly reduced by using average pooling and max pooling, creating two-dimensional feature maps of \( X_A, X_M \in \mathbb{R}^{W \times H} \), respectively. Then, a network layer (e.g., a normal convolutional layer), described by \( conv() \) is applied and followed by a Sigmoid function. The SA layer is formulated as Eq. 4:

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

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, creating a robust guidance for the network to process important regions of a certain feature map. This attention mechanism can be described by formulas: Eq. 5 and Eq. 6:

\[
X' = f_{CA}(X)
\]

\[
f_{CBAM} = f_{SA}(X')
\]

where \( f_{CA} \) and \( f_{SA} \) are from Eq. 3 and Eq. 4.

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

\[
Q = X \cdot W_q
\]

\[
K = X \cdot W_k
\]

\[
V = X \cdot W_v
\]

where \( W_q, W_k, W_v \) are weight matrices. Then, the output of an attention head can be calculated using Eq. 10:

\[
g_n = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

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

As each attention head learns a different set of weight matrices \( W_q, W_k, W_v \), 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 \( X \) is described by

\[
\sum_{n=1}^{N} g_n \odot X
\]

### 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 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 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.

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.

3.3 Apply different 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 example, EfficientB0 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 based on the formulations mentioned in Section 2.2. For the Multihead attention scheme, we propose a Multihead attention based layer as shown in Figure 5. In particular, given an input feature map X with a size of $[W \times H \times C]$ where $W$, $H$, and $C$ presents width, height, and channel dimensions, we reduce the size of feature map $X$ across three dimensions using both max and average pooling layers. We then generate 3 two-dimensional feature maps, likely matrices of: $[W \times H]$, $[H \times C]$, $[W \times C]$. Next we feed all generated matrices into the Multihead attention to obtain attention score matrices. Then, we reshape the attention score matrices into the sizes of $[W \times H \times 1]$, $[1 \times H \times C]$, $[W \times 1 \times C]$ respectively and element-wise multiply each of them with the input feature map $X$. Finally, we conduct an average of three results of multiplications, generate the output tensor $Y$ with the size of $[W \times H \times C]$ which is same size as the input feature map $X$. 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 $[H \times C]$), $[W \times C]$) and spatial feature (feature map with size of $[W \times H]$) are focused, which helps 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 $X$ to an output feature map $Y$ and retains the size of the input feature map $X$, we then apply a global average pooling layer after each attention layer to scale down the feature maps to one-dimensional feature maps, likely vectors. Finally, these vectors are added together and feed into 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, consisting of 31,500 remote sensing images. The remote sensing images are separated into 45 scene classes, each of which comprises 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-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.

$$Loss_{KL}(\Theta) = \sum_{n=1}^{N} y_n \log \left( \frac{y_n}{\hat{y}_n} \right) + \frac{\lambda}{2} ||\Theta||^2_2,$$ (12)
Table 1: 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)/Memory (MB) |
|------------------|--------------|----------------------------|
| MobileNet        | 90.2         | 3.7/14.8                   |
| MobileNetV2      | 90.9         | 2.9/11.6                   |
| NASNetMobile     | 91.7         | 4.8/19.2                   |
| EfficientNetB0   | 92.0         | 4.6/18.4                   |

Table 2: 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 Multthead |
|-----------------|----------|---------|--------------------|
| Accuracy (%)    | 92.1     | 92.3    | 93.8               |
| Parameters (M)  | 10.4     | 6.7     | 9.4                |
| Memory (MB)     | 41.6     | 26.8    | 37.6               |

where Θ presents trainable parameters, the constant λ is empirically set to 0.0001, the batch size N is set to 60, ŷt and ŷt 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 epochs. While the first 50 epochs uses the learning rate of 0.0001 and all data augmentation methods mentioned in Section 3.1, the remaining 10 epochs use 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 1, the proposed RSIC system using the transfer learning technique and EfficientNetB0 and 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/18.8 MB) is also smaller than NASNetMobile (4.8 M/19.2 MB), 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 Multthead attention. In this experiment, only the feature maps which are extracted from the final three convolutional blocks (block 5 to block 7) in the EfficientNetB0 backbone are used. As Table 2 shows, applying attention layers helps to improve the system performance by 0.1%, 0.3%, and 1.8% with SE, CBAM, and the proposed Multthead attention, respectively.

As the proposed Multthead attention layer outperforms SE and CBAM layers, we then evaluate the proposed Multthead attention with a different number of feature maps. As the results are shown in Table 3, using three feature maps still achieves the best performance. Regarding the model complexity, using the proposed Multthead attention layer with three feature maps increases the model footprint from 4.6 M to 9.4 M, which leads to occupy from 18.4 MB to 37.6 MB memory (i.e. 1 trainable parameter is presented by 32 bit in floating point format). To meet the constrain of maximum memory occupation of 20 MB on application device, we apply the quantization technique which helps to reduce the model complexity to 9.4 MB (i.e. The quantization technique helps 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 memory 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 Multthead attention layer for three feature maps and the quantization technique, we achieve a low-complexity RISC model (9.4 M trainable parameters and 9.4 MB memory occupation). 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 footprint (9.4 M / 9.4 MB) compared with the best Transformer based model (46.3 M / 185.2 MB).

Table 3: Performance (Acc.%) of EfficientNetB0 with the proposed Multthead attention applied for feature maps extracted from different convolutional blocks on the benchmark NWPU-RESISC45 dataset using 20%-80% splitting settings.

| Convolutional Blocks | Accuracy (%) | Parameters (M)/Memory (MB) |
|----------------------|--------------|----------------------------|
| Blocks 7             | 93.1         | 6.0/24                     |
| Blocks 6 to 7        | 93.0         | 7.5/30                     |
| Blocks 5 to 7        | 93.8         | 9.4/37.6                   |
| Blocks 4 to 7        | 92.8         | 11.2/44.8                  |
| Blocks 3 to 7        | 92.5         | 13.3/53.2                  |

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) (35.99 Mb) [43]   | 89.4         | 91.7         |
| MG-CAP (Bilinear) (55.99 Mb) [43]| 89.4         | 91.0         |
| MG-CAP (Squ-E) (55.99 Mb) [43]   | 90.8         | 93.0         |
| EfficientNetB0-aux (≈ 5.3 Mb) [4] | 90.0     | 92.9         |
| EfficientNetB0-aux (≈ 13 Mb) [2] | 91.1     | 93.8         |
| VGG-16 + MTL (≈ 138.4 Mb) [62]   | 91.5         |              |
| ResNeXt-50 - MTL (≈ 25.2 Mb) [62]| 93.8         |              |
| ResNeXt-101 - MTL (≈ 88.79 Mb) [62]| 91.9     | 94.2         |
| SE-MDPMNet (5.17 Mb) [44]        | 91.8         | 94.1         |
| LGRIN (4.63 Mb) [49]             | 91.9         | 94.4         |
| Transformer (46.3 Mb) [57]       | 93.1         | 95.6         |
| Our systems (9.4 Mb/9.4 MB)      | 91.0         | 93.8         |
5 CONCLUSION
This paper has presented a deep learning model for remote sensing image classification (RSIC). By conducting extensive experiments, we indicate that applying multiple techniques of transfer learning, Multilead 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] Salkat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Kurki, and Ramakrishna Nemani. 2015. Deeppsat: a learning framework for satellite imagery. In Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. 1–10.
[2] Yakoob Bazi, Mohamad M Al Rahhal, Haikel Alhichri, and Naif Alajlan. 2019. Simple yet effective fine-tuning of deep CNNs using an auxiliary classifying loss for remote sensing scene classification. Remote Sensing, 11, 24 (2019), 2908.
[3] Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing 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 Elmanani, Mohamed Anis Loghmari, and Mohamed Saber Naceur. 2016. A new convolutional 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 Elmanani, Mohamed Anis Loghmari, and Mohamed Saber Naceur. 2013. Support vector machine for remote sensing image classification. In Proceedings of 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] Yiyuan Guo, Jinsheng Li, 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, Junqang Yang, Longguang Wang, Yingqian Wang, and Wei An. 2021. Gated recurrent multilattention network for VRH remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 60 (2021), 1–13.
[18] Fengpeng Li, Ruoyi Feng, Wei Han, and Linhe 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 Mehmoond, Ahsan Shahzad, Bushra Zafar, Amna Shabir, and Nourman Ali. 2022. Remote sensing image classification: A comprehensive review and applications. Mathematical Problems in Engineering 2022 (2022).
[20] Rodrigo Minetto, Mauricio 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] Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. 2021. Mind your weight: A survey of data-centric models for social network analysis. Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems. 1–10.
