Remote Sensing Image Classification using Transfer Learning and Attention Based Deep Neural Network

Lam Pham¹, Khoa Tran², Dat Ngo³, Jasmin Lampert¹, and Alexander Schindler¹

¹Center for Digital Safety & Security, Austrian Institute of Technology, Austria.
²University of Science and Technology, University of Danang, Viet Nam.
³School of Computer Science and Electronic Engineering, University of Essex, UK.

ABSTRACT

The task of remote sensing image scene classification (RSISC), which aims at classifying remote sensing images into groups of semantic categories based on their contents, has assumed an important role in a wide range of applications such as urban planning, natural hazards detection, environmental monitoring, vegetation mapping or geospatial object detection. During the past years, the research community focusing on RSISC tasks has shown significant effort to publish diverse datasets as well as to propose different approaches. Recently, almost all proposed RSISC systems are based on deep learning models, which proves powerful and outperform traditional approaches using image processing and machine learning. In this paper, we also leverage the power of deep learning technologies, evaluate a variety of deep neural network architectures and indicate main factors affecting the performance of a RSISC system. Given the comprehensive analysis, we propose a deep learning based framework for RSISC, which makes use of a transfer learning technique and a multihead attention scheme. The proposed deep learning framework is evaluated on the NWPU-RESISC45 benchmark dataset and achieves a classification accuracy of up to 92.6% and 94.7% with two official data split suggestions (10% and 20% of entire the NWPU-RESISC45 dataset for training). The achieved results are very competitive and show potential for real-life applications.

Keywords: Convolutional Neural Network (CNN), Transfer Learning, Attention, Remote Sensing Image, Data Augmentation

1. INTRODUCTION

Remote sensing image scene classification (RSISC), which was already mentioned in,¹,² has recently attracted much research attention for leveraging the power of deep learning techniques. Indeed, while handcrafted features using image processing techniques such as color histogram,¹,³ Gabor-based texture features,¹ SIFT description²,³ and traditional machine learning models for classification such as Support Vector Machine (SVM),¹,² Gaussian Mixture based Clustering or Classification³,⁴ were widely applied before, recent publications for RSISC have presented a wide range of deep learning models, mainly based on deep convolutional neural network architectures. Generally, to aggregate global features in a remote sensing image, recently proposed models make use of the convolution operation, stack multiple convolutional layers to make the networks deeper and easier to train. For instance, ResNet50 and ResNet101 architectures were explored in,⁵ presenting very competitive results on different benchmark datasets of NWPU-RESISC45,⁶ AID,³ UC-Merced,¹ and WHU-RS19.⁷ Similarly, authors in⁸ deployed EffectiveNet-based models while VGG-based network architectures such as AlexNet, VGG-16, or GoogLeNet were fine-tuned in.⁹

It can be seen that recent RSISC systems make an effort to explore or fine-tune specific neural network architectures, but have not provided a detailed analysis of factors affecting the network performances or a comprehensive comparison among different network architectures. This inspired us to conduct extensive experiments in this paper and to present three main contributions:

Lam Pham and Khoa Tran are main and equal contribution into the paper.
2. PROPOSED DEEP LEARNING FRAMEWORKS

As Figure 1 shows, the high-level architecture of our proposed deep learning frameworks comprises two main steps: (1) data augmentation to make the input images diverse and (2) back-end deep neural networks for classification.

2.1 Data augmentation methods

In this paper, we apply four methods of data augmentation: Image Rotation (IR),\textsuperscript{10} Random Cropping (RC),\textsuperscript{10} Random Erasing (RE),\textsuperscript{11} and Mixup (Mi).\textsuperscript{12,13} In particular, all images in the target dataset are rotated with angles set to 90, 180, and 270, referred to as Image Rotation (IR). By using this method, the dataset increases by a factor of four. Then, the images are randomly grouped into batches with a batch size set to 32. For each batch, the images are randomly cropped with a reduction of 10 pixels on both of width and height dimensions, referred to as Random Cropping (RC). Then, 20 random pixels on both width and height dimensions of each image are erased, referred to as Random Erasing (RE). Finally, the images are mixed together with random ratios, referred to as Mixup (Mi). As we use both Uniform or Beta distributions to generate the mixup ratios, the batch size increases from 32 to 96. Batches of 96 images are then fed into the deep neural networks for classification. As Random Cropping (RC),\textsuperscript{10} Random Erasing (RE),\textsuperscript{11} and Mixup (Mi)\textsuperscript{12,13} are used on batches of images during the training process, they are referred to as on-line data augmentation. Meanwhile, Image Rotation (IR)\textsuperscript{10} is referred to as off-line data augmentation as this method is applied on all training datasets before the training process.

2.2 Proposed deep neural networks for classification

As the high-level architecture of the proposed deep learning frameworks is shown in Figure 1, the back-end deep neural network, which presents a convolutional-based architecture, can be separated into two main parts: (1) The convolutional-based backbone (CNN-based backbone), which is performed by trunks of convolutional-based layers, helps to transfer the input images to condensed feature map; and (2) the multilayer perceptron based classification (MLP-based classification) for classifying the feature map extracted from CNN-based backbone into certain categories.

In this paper, the proposed MLP-based classification is performed by a fully connected layer (FC(channel number)), a rectified linear unit (ReLU),\textsuperscript{14} and a dropout (Dr(drop ratio)),\textsuperscript{15} and softmax layer, as described in Table 1. Notably, the number of channels as the second fully connected layer is set to $C$ that presents the number of categories of the target RICSC dataset (i.e., for an example, $C$ is set to 45 for the NWPU-RESISC45
Meanwhile, to find the best architecture for the CNN-based backbone, we reuse and evaluate a wide range of benchmark deep neural networks. In particular, we firstly evaluate residual based network architectures of ResNet152, ResNet152V2, DenseNet121, DenseNet201, which present a deep trunk of convolutional layers, showing a scale up of convolutional layers by depth. We then evaluate InceptionV3 architectures, which scale up convolutional layers by width with multiple kernels. We also evaluate InceptionResNetV2 network which shows a hybrid form between residual based and inception based architectures. Recently, EfficientNet architectures have been developed to aim at uniformly scaling all dimensions of depth, width, and image resolution. We therefore evaluate two forms of EfficientNet architectures, EfficientNetB0 and EfficientNetB4, in this paper. These benchmark networks of ResNet152, ResNet152V2, InceptionV3, InceptionResNetV2, DenseNet121, DenseNet201, EfficientNetB0, EfficientNetB4 are available in the Keras library.

By reusing these benchmark networks, we propose two training strategies: direct training and transfer learning as shown in Figure 2. In the direct training strategy as shown in the upper part of Figure 2, we reuse the first layer to the global pooling layer of these benchmark networks, referred to as the benchmark-network backbone, to perform the proposed CNN-based backbone. All trainable parameters of the proposed deep neural networks in the direct training strategy (i.e., all trainable parameters of both the CNN-based backbone and the MLP-based classification) are initialized with random values of mean 0 and variance 0.1. For the transfer learning strategy as shown in the lower part in Figure 2, the benchmark network architectures were trained with the large-scale ImageNet1K dataset in advance. Then, trainable parameters of the benchmark-network backbone are transferred to the CNN-based backbone. In other words, only trainable parameters of the MLP-based classification are initialized, while those of the CNN-based backbone are reused from the pre-trained benchmark network architectures in the transfer learning strategy. While the direct training strategy uses a learning rate of 0.0001, a lower learning rate of 0.00001 is set for the transfer learning strategy.

2.3 Proposed a multihead attention based network layer

Taking into consideration that traditional global pooling layers may not perform as well as a global max or average value across width and height dimensions and that they cannot fully represent an entire two-dimensional feature dataset). Meanwhile, to find the best architecture for the CNN-based backbone, we reuse and evaluate a wide range of benchmark deep neural networks. In particular, we firstly evaluate residual based network architectures of ResNet152, ResNet152V2, DenseNet121, DenseNet201, which present a deep trunk of convolutional layers, showing a scale up of convolutional layers by depth. We then evaluate InceptionV3 architectures, which scale up convolutional layers by width with multiple kernels. We also evaluate InceptionResNetV2 network which shows a hybrid form between residual based and inception based architectures. Recently, EfficientNet architectures have been developed to aim at uniformly scaling all dimensions of depth, width, and image resolution. We therefore evaluate two forms of EfficientNet architectures, EfficientNetB0 and EfficientNetB4, in this paper. These benchmark networks of ResNet152, ResNet152V2, InceptionV3, InceptionResNetV2, DenseNet121, DenseNet201, EfficientNetB0, EfficientNetB4 are available in the Keras library.

By reusing these benchmark networks, we propose two training strategies: direct training and transfer learning as shown in Figure 2. In the direct training strategy as shown in the upper part of Figure 2, we reuse the first layer to the global pooling layer of these benchmark networks, referred to as the benchmark-network backbone, to perform the proposed CNN-based backbone. All trainable parameters of the proposed deep neural networks in the direct training strategy (i.e., all trainable parameters of both the CNN-based backbone and the MLP-based classification) are initialized with random values of mean 0 and variance 0.1. For the transfer learning strategy as shown in the lower part in Figure 2, the benchmark network architectures were trained with the large-scale ImageNet1K dataset in advance. Then, trainable parameters of the benchmark-network backbone are transferred to the CNN-based backbone. In other words, only trainable parameters of the MLP-based classification are initialized, while those of the CNN-based backbone are reused from the pre-trained benchmark network architectures in the transfer learning strategy. While the direct training strategy uses a learning rate of 0.0001, a lower learning rate of 0.00001 is set for the transfer learning strategy.

2.3 Proposed a multihead attention based network layer

Taking into consideration that traditional global pooling layers may not perform as well as a global max or average value across width and height dimensions and that they cannot fully represent an entire two-dimensional feature

Figure 2. Direct training and transfer learning strategies by reusing the network backbone from benchmark deep neural network architectures.

Table 1. The proposed MLP-based classification

| Setting Layers | Outputs |
|----------------|---------|
| FC(4096) - ReLU - Dr(0.2) | 4096 |
| FC(C) - Softmax | C |
map, we therefore evaluate whether applying the attention technique can help to enhance the performance. In particular, we replace the global pooling layer in the CNN-based backbone by a multihead attention based layer as shown in Figure 3. The proposed layer is developed to learn distinct features across the width and height dimensions using a multihead attention scheme. To this end, let us consider the shape of an input tensor of the proposed multihead attention based layer as \([B \times W \times H \times C]\), where \(B\), \(W\), \(H\), and \(C\) are the batch size, width dimension, height dimension and channel dimension, respectively. We then reduce either the width or the height dimension of the input tensor using average or max pooling layers, generating tensors of \([B \times W \times C]\) or \([B \times H \times C]\) with two independent data streams. Next, the multihead attention scheme is applied on two data streams, each of which focuses on learning distinct features across either the width dimension (i.e., the upper data stream in Figure 3) or the height dimension (i.e., the lower data stream in Figure 3). Notably, we empirically set the number of heads to 16, set the key dimension to 128, and retain the shape of the input tensors. Then, average or max pooling layers are again applied to reduce the width or height dimensions, achieving tensors of \([B \times C]\) on each data stream. Finally, two data streams are concatenated to generate a tensor of \([B \times 2C]\) before transferring them into a MLP-based classification.

2.4 Apply an ensemble to enhance the performance

Since the predicted probabilities obtained from individual deep neural network architectures can complement each other and a fusion of these predicted probabilities can help to improve the performance, we propose an ensemble of predicted probabilities in this paper, referred to as PROD late fusion. Let us consider predicted probability of each deep neural network as \(\bar{p}_n = (\bar{p}_{n1}, \bar{p}_{n2}, ..., \bar{p}_{nC})\), where \(C\) is the category number and the \(n^{th}\) out of \(N\) networks evaluated, the predicted probability after PROD fusion \(p_{prod} = (\bar{p}_{1}, \bar{p}_{2}, ..., \bar{p}_{C})\) is obtained by:

\[
\bar{p}_c = \frac{1}{N} \prod_{n=1}^{N} \bar{p}_{nc} \quad for \quad 1 \leq n \leq N
\]  

Finally, the predicted label \(\hat{y}\) is determined by

\[
\hat{y} = \arg\max(\bar{p}_{1}, \bar{p}_{2}, ..., \bar{p}_{C})
\]  

3. EXPERIMENTS AND RESULTS

3.1 Dataset and evaluation metric

In this paper, we evaluate the NWPU-RESISC45 benchmark dataset, which consists of 31,500 remote sensing images collected from more than 100 countries and regions all over the world. 31,500 images are separated into 45 scene classes, each of which comprises 700 images in RGB color format with a size of 256 × 256 pixels. Compared to the other benchmark datasets, AID, UC-Merced and WHU-RS19, the NWPU-RESISC45 dataset presents the larger number of categories and the larger number of images per category.
Table 2. Performance (Acc.(/%)/Training Epoches) comparison among deep learning frameworks with direct training (DR) and transfer learning (TL), with or without using data augmentation (Aug.) and attention (Att.) on the benchmark NWPU-RESISC45 dataset with 20%-80% data splitting setting.

| Configuration/Networks | DenseNet121 | DenseNet201 | InceptionV3 | InceptionResNetV2 | EfficientNetB0 | EfficientNetB4 |
|------------------------|-------------|-------------|-------------|-------------------|----------------|----------------|
| DT                     | 49.7/146    | 45.5/146    | 64.7/58     | 49.8/132          | 43.5/132       | 43.5/132       |
| TL                     | 56.7/83     | 45.5/146    | 84.1/31     | 58.2/76           | 82.4/49        | 81.5/49        |
| TL & Aug.              | 89.1/16     | 91.8/31     | 91.7/23     | 89.6/18           | 91.3/38        | 91.8/22        |
| TL & Aug. & Att.       | 93.6/10     | 92.9/13     | 90.9/17     | 90.5/11           | 92.9/26        | 92.8/12        |

Table 3. Performance comparison using late fusion of DenseNet201 and EfficientNetB4, max and average pooling layers inside the proposed attention on the benchmark NWPU-RESISC45 dataset with 20%-80% data splitting setting.

| Pooling layers | DenseNet201 | EfficientNetB4 | DenseNet201 + EfficientNetB4 | DenseNet201 + EfficientNetB4 |
|----------------|-------------|----------------|-------------------------------|-------------------------------|
| Max + Average  | 93.6        | 93.1           | 94.3                          | 94.7                          |

To compare with the state-of-the-art systems, we follow the original settings in, and then split the NWPU-RESISC45 dataset into Training-Testing subsets with two different ratios: 20%-80% and 10%-90%. We also follow the original paper by using accuracy (Acc.%) as the metric to evaluate our proposed systems in this paper.

3.2 Experimental settings

As we apply the Mixup data augmentation, labels are no longer in one-hot encoding format. Instead, we use a Kullback-Leibler divergence (KL) loss:

$$\text{Loss}_{KL}(\Theta) = \sum_{n=1}^{N} y_n \log \left( \frac{y_n}{\hat{y}_n} \right) + \frac{\lambda}{2} \| \Theta \|^2,$$

where $\Theta$ are trainable parameters and the constant $\lambda$ is set initially to 0.0001. Additionally, the batch size $N$, $y_i$ and $\hat{y}_i$ denote expected and predicted results. All deep learning networks proposed are constructed with Tensorflow, and the experiments were performed on two 24 GB Nvidia Titan-GPUs for both training and inference jobs.

3.3 Performance comparison between direct training and transfer learning approaches

We firstly compare the direct training (DT) and transfer learning (TL) strategies without using both data augmentation methods and the proposed attention layer. As results are shown in Table 2, we can see that the networks with the transfer learning strategy significantly outperform those using direct training. Almost all proposed deep learning networks using the transfer learning strategy achieve more than 80% with an exception of ResNet152V2. The transfer learning strategy also helps to reduce the training time, especially for the large networks, i.e. ResNet152, ResNet152V2, or EfficientNetB4. Comparing different deep neural networks, DenseNet based architectures outperform the others regarding using the transfer learning strategy, recording classification accuracy scores of 88.7% and 88.1% for DenseNet121 and DenseNet201, respectively.

When we apply different data augmentation methods, the performance of EfficientNetB4 significantly improves by 10%, while the other networks show an average improvement of 3%. By using a transfer learning strategy and data augmentation methods, the EfficientNet and DenseNet based architectures are competitive and outperform ResNet and Inception based networks. Although training times (i.e. the epoch numbers) are significantly reduced, each training epoch needs more time to finish as the data augmentation methods increase the number of input data.

Next, we evaluate the transfer learning strategy using both data augmentation methods and the proposed attention layer (i.e., notably, the average pooling layer is used in the proposed attention layer). As the final line in Table 2 shows, InceptionV3 and DenseNet, EfficientNet based architectures improve by an average score of 1%. Moreover, an improvement of more than 3% is achieved with InceptionResNetV2, but an insignificant improvement is observed for ResNet based deep neural networks. Comparing performances among networks,
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 |
|-----------------------------------|--------------|--------------|
| APDC-Net                          | 85.9         | 87.8         |
| D-CNN with GoogLeNet              | 86.9         | 90.5         |
| VGG-16 + MTL                      | -            | 91.5         |
| MG-CAP (Log-E)                    | 89.4         | 91.7         |
| MG-CAP (Bilinear)                 | 89.4         | 93.0         |
| EfficientNet-B0-aux                | 90.0         | -            |
| MG-CAP (Sqrt-E)                   | 90.8         | 93.0         |
| ResNet-50 + MTL                   | -            | 93.8         |
| ResNet-101 + MTL                  | 91.9         | 94.2         |
| EfficientNet-B3-aux               | 91.1         | -            |
| SE-MDPMNet                        | 91.8         | 94.1         |
| Xu's method                       | 91.9         | 94.4         |
| Our systems                       | 92.6         | 94.7         |

DenseNet and EfficientNet based architectures are competitive and outperform the others, recording an average score of 92.8%. Applying an attention layer also helps to reduce training time, especially for the large networks of DenseNet201 and EfficientNetB4 with 8 and 12 training epochs.

As DenseNet201 and EfficientNetB4 achieve the best performances compared to the other architectures, we conduct PROD late fusion of the predicted probabilities obtained from these networks as mentioned in Section 2.4. Additionally, as the average or max pooling layer used in the proposed attention focuses on either max or average of feature map across the width or height dimension, we evaluate whether the average and max pooling layers can complement. In particular, PROD late fusion of predicted probabilities from DenseNet201 or EfficientNetB4 using either max or average pooling layer are evaluated. As shown in Table 3, we can see that ensembles of max and average pooling layers or ensembles of DenseNet201 and EfficientNetB4 network architectures can help to improve the performance. When we conduct ensembles of four individual models: DenseNet201 with max pooling layer, DenseNet201 with average pooling layer, EfficientNetB4 with max pooling layer, and EfficientNetB4 with average pooling layer, we can achieve the best classification accuracy score of 94.7%.

3.4 Performance comparison to the state-of-the-art

In Table 4, which shows the performance comparison with the state-of-the-art systems, we can see that our best models are very competitive, recording accuracy scores of 92.6% and 94.7% with training ratios of 10% and 20%, respectively. These results show potential for applying RSISC tasks to real-life applications.

4. CONCLUSION

This paper has presented an exploration of various deep learning models for remote sensing image classification (RSISC). By conducting extensive experiments, we indicate that applying multiple techniques of transfer learning, data augmentation, attention scheme, and ensemble to DenseNet and EfficientNet based deep learning frameworks is effective to achieve a high-performance RSISC system.

Our further research will focus on developing a low-complexity RSISC system, which can be adapted to a wide range of applications on edge devices.

ACKNOWLEDGMENTS

The research leading to this publication was partially carried out within the gAia project. The gAia project is funded by the KIRAS program of the Austrian Research Promotion Agency (FFG) and the Federal Ministry of Agriculture, Regions and Tourism (BMLRT) under grant no. FO999886369.
REFERENCES

[1] Yang, Y. and Newsam, S., “Bag-of-visual-words and spatial extensions for land-use classification,” in [18th SIGSPATIAL international conference on advances in geographic information systems], 270–279 (2010).

[2] Yang, Y. and Newsam, S., “Comparing sift descriptors and gabor texture features for classification of remote sensed imagery,” in [15th IEEE International Conference on Image Processing], 1852–1855 (2008).

[3] Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., and Lu, X., “Aid: A benchmark data set for performance evaluation of aerial scene classification,” IEEE Transactions on Geoscience and Remote Sensing 55(7), 3965–3981 (2017).

[4] Kusumaningrum, R., Wei, H., Manurung, R., and Murni, A., “Integrated visual vocabulary in latent dirichlet allocation–based scene classification for ikonos image,” Journal of Applied Remote Sensing 8(1), 083690 (2014).

[5] Zhao, Z., Luo, Z., Li, J., Chen, C., and Piao, Y., “When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework,” Remote Sensing 12(20), 3276 (2020).

[6] Cheng, G., Han, J., and Lu, X., “Remote sensing image scene classification: Benchmark and state of the art,” Proceedings of the IEEE 105(10), 1865–1883 (2017).

[7] Xia, G.-S., Yang, W., Delon, J., Gousseau, Y., Sun, H., and Mâître, H., “Structural high-resolution satellite image indexing,” in [100 Years ISPRS - Advancing Remote Sensing Science], (2010).

[8] Bazi, Y., Al Rahhal, M. M., Alhichri, H., and Alajlan, N., “Simple yet effective fine-tuning of deep cnns using an auxiliary classification loss for remote sensing scene classification,” Remote Sensing 11(24), 2908 (2019).

[9] Cheng, G., Han, J., and Lu, X., “Remote sensing image scene classification: Benchmark and state of the art,” Proceedings of the IEEE 105(10), 1865–1883 (2017).

[10] Shorten, C. and Khoshgoftaar, T. M., “A survey on image data augmentation for deep learning,” Journal of big data 6(1), 1–48 (2019).

[11] Takahashi, R., Matsubara, T., and Uehara, K., “Data augmentation using random image cropping and patching for deep cnns,” IEEE Transactions on Circuits and Systems for Video Technology 30(9), 2917–2931 (2020).

[12] Xu, K., Feng, D., Mi, H., Zhu, B., Wang, D., Zhang, L., Cai, H., and Liu, S., “Mixup-based acoustic scene classification using multi-channel convolutional neural network,” in [Pacific Rim Conference on Multimedia], 14–23 (2018).

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

[14] Nair, V. and Hinton, G. E., “Rectified linear units improve restricted boltzmann machines,” in [ICML], (2010).

[15] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., “Dropout: a simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research 15(1), 1929–1958 (2014).

[16] Tan, M. and Le, Q., “Efficientnet: Rethinking model scaling for convolutional neural networks,” in [International conference on machine learning], 6105–6114 (2019).

[17] et al., F. C., “Keras library.” https://keras.io (2015).

[18] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., “Imagenet: A large-scale hierarchical image database,” in [IEEE Conference on Computer Vision and Pattern Recognition], 248–255 (2009).

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I., “Attention is all you need,” in [Advances in Neural Information Processing Systems], (2017).

[20] Minetto, R., Segundo, M. P., and Sarkar, S., “Hydra: An ensemble of convolutional neural networks for geospatial land classification,” IEEE Transactions on Geoscience and Remote Sensing 57(9), 6530–6541 (2019).

[21] Pham, L., Ngo, D., Nguyen, P. X., Hoang, T., and Schindler, A., “An audio-visual dataset and deep learning frameworks for crowded scene classification,” arXiv preprint arXiv:2112.09172 (2021).
[22] Pham, L., Schindler, A., Schutz, M., Lampert, J., Schlarb, S., and King, R., “Deep learning frameworks applied for audio-visual scene classification,” in [Data Science–Analytics and Applications], 39–44 (2022).

[23] Kullback, S. and Leibler, R. A., “On information and sufficiency,” The annals of mathematical statistics 22(1), 79–86 (1951).

[24] Bi, Q., Qin, K., Zhang, H., Xie, J., Li, Z., and Xu, K., “Apdc-net: Attention pooling-based convolutional network for aerial scene classification,” IEEE Geoscience and Remote Sensing Letters 17(9), 1603–1607 (2020).

[25] Cheng, G., Yang, C., Yao, X., Guo, L., and Han, J., “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns,” IEEE transactions on geoscience and remote sensing 56(5), 2811–2821 (2018).

[26] Wang, S., Guan, Y., and Shao, L., “Multi-granularity canonical appearance pooling for remote sensing scene classification,” IEEE Transactions on Image Processing 29, 5396–5407 (2020).

[27] Zhang, B., Zhang, Y., and Wang, S., “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), 2636–2653 (2019).

[28] Xu, C., Zhu, G., and Shu, J., “A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification,” IEEE Transactions on Geoscience and Remote Sensing 60, 1–15 (2021).