

A Light-weight Deep Learning Model for Remote Sensing Image Classification

Lam Pham*

Austrian Institute of Technology
Vienna, Austria
lam.pham@ait.ac.at

Cam Le*

HCM University of Technology
HCM, VietNam
cam.levt123@hcmut.edu.vn

Dat Ngo

University of Essex
Colchester, UK
dn22678@essex.ac.uk

Anh Nguyen

FPT Soft Company
HCM, VietNam
AnhNTN34@fsoft.com.vn

Jasmin Lampert

Austrian Institute of Technology
Vienna, Austria
Jasmin.Lampert@ait.ac.at

Alexander Schindler

Austrian Institute of Technology
Vienna, Austria
Alexander.Schindler@ait.ac.at

Ian McLoughlin

Singapore Institute of Technology
Singapore
ian.mcloughlin@singaporetech.edu.sg

Abstract—In this paper, we present a high-performance and light-weight deep learning model for Remote Sensing Image Classification (RSIC), the task of identifying the aerial scene of a remote sensing image. To this end, we first evaluate various benchmark convolutional neural network (CNN) architectures: MobileNet V1/V2, ResNet 50/151V2, InceptionV3/InceptionResNetV2, EfficientNet B0/B7, DenseNet 121/201, ConNeXt Tiny/Large. Then, the best performing models are selected to train a compact model in a teacher-student arrangement. The knowledge distillation from the teacher aims to achieve high performance with significantly reduced complexity. By conducting extensive experiments on the NWPU-RESISC45 benchmark, our proposed teacher-student models outperforms the state-of-the-art systems, and has potential to be applied on a wide range of edge devices.

Index Terms—Teacher-student model, convolutional neural network (CNN), data augmentation, high-level features.

I. INTRODUCTION

Remote sensing image classification (RSIC) is a core task for a range of real-world applications including land use classification, natural hazard assessment [1], scene-driven geospatial object detection [2], and environmental monitoring [3]. The task has therefore drawn much attention from the research community in recent years, including in the area of datasets and benchmarks. The earliest RSIC dataset, UCM [4], was proposed in 2010. Subsequently, more challenging RSIC datasets have been published, such as NWPU VHR-10 (2014) [5], SAT6 (2015) [6], SIRI-WHU (2015) [7], AID (2017) [8], OPTIMAL (2018) [9], NWPU-RESISC45 (2017) [10], etc. Among these published datasets, NWPU-RESISC45 has the largest number of classes, comprising 45 image scenes, each of which is represented by 700 remote sensing images. Additionally, a wide range of classification models have been published for RSIC tasks. Early systems used conventional image processing techniques such as Texture Descriptors (TD) [11], Local binary patterns (LBP) [12], Color Histogram (CH), Histogram of Oriented Gradient (HOG) [13], Scale-Invariant

Feature Transformation (SIFT) [14] to extract hand-crafted features. Then, these features were classified by traditional machine learning based models such as Support Vector Machine (SVM) [10], [15], Gaussian Mixture Model (GMM) [16], etc. More recently proposed RSIC systems leveraged deep learning based network architectures, which have proven to be more effective compared to traditional machine learning methods [17], [18]. Most deep learning based systems for RSIC make use of Convolutional Neural Network (CNN) based architectures such as ResNet [19], DenseNet [20], EfficientNet [21] or Transformer [22]. Although deep learning based RSIC systems have demonstrated the potential for very good performance [23], these network architectures involve large footprint models with a high number of trainable parameters [23]. This causes challenges to apply such deep learning based RSIC models within edge devices [24]. In this paper, we aim to develop a low footprint RSIC model which is capable of achieving high-performance by leveraging the strength of advanced high complexity models to achieve cutting-edge performance. The resulting distilled student architecture achieves a model size reduction of 98% at the cost of a 1.4% relative drop in performance. Our main contributions are as follows:

(a) A mechanism to combine individual high-performing CNN-based networks trained on the RSIC task, to inform a single robust teacher network. Given the teacher, we apply a teacher-student scheme to train the student. Using knowledge distillation from the teacher, the student not only performs well but is also a low complexity model. In this paper, we propose a constraint of maximum 5 million trainable parameters for a low-complexity RSIC model. This is consistent with the capability of typical edge devices. (b) We evaluate our proposed teacher and student models on the NWPU-RESISC45 benchmark [10]. Results reveal that the proposed models outperform state-of-the-art systems with or without considering the issue of complexity – demonstrating the ability of the technique to enable implementation on a range of edge devices.

(*) Lam Pham and Cam Le made equal contribution to this paper.

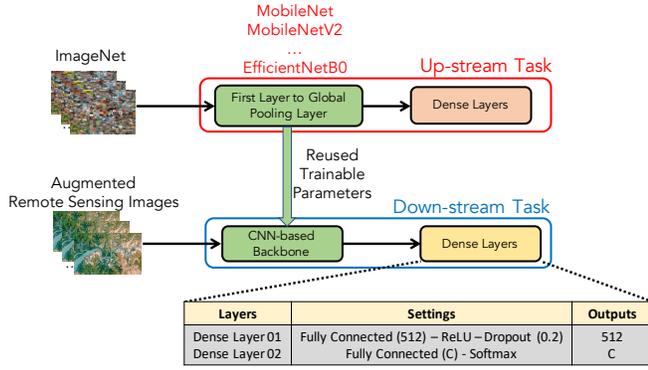


Fig. 1. Evaluation of benchmark networks using the transfer learning technique.

II. THE THREE-PHASE PROCESS TO DEVELOP AND ACHIEVE A HIGH-PERFORMANCE AND LOW-COMPLEXITY RSIC MODEL

In this section, we describe the methods employed to achieve a high-performance and low-complexity RSIC model, which leverages a teacher-student arrangement [25]. In particular, the process comprises three main phases:

- Phase I: We first evaluate a wide range of benchmark convolution neural network (CNN) based architectures. Then, we select which networks (i.e. the best performance models) to use for developing the teacher model, and which network is used for the student model (i.e. the student model not only performs well but also presents a low footprint).
- Phase II: In this phase, the best performance models from Phase I are used to develop the teacher. After training the proposed teacher, the feature maps at the next to last dense layer of the teacher are extracted. The extracted feature maps are referred to as high-level features.
- Phase III: Finally, the student network, which selected in Phase I, is trained with the high-level features (i.e. via knowledge distillation from the teacher) to achieve a high-performance and low-complexity RSIC model.

A. Phase I: Evaluate the benchmark neural networks to select high-performance networks for the teacher and student

We assessed various convolutional neural network (CNN) based architectures for both the teacher and the student models by evaluating twelve different benchmark deep convolutional neural networks: MobileNet, MobileNetV2, ResNet50, Resnet151V2, InceptionV3, InceptionResNetV2, DenseNet121, DenseNet201, EfficientNetB0, EfficientNetB7, ConvNeXtTiny, and ConvNeXtLarge, all available in the Keras library [26]. As the top of Figure 1 shows, the benchmark networks are first trained with the ImageNet dataset [27], referred to as the up-stream task. Then, the first layer to the global pooling layer of these pre-trained networks are extracted and combined with a Dense Layers block to perform the down-stream RSIC task as shown at the bottom of Figure 1. In other words, we apply a transfer learning method in which the first

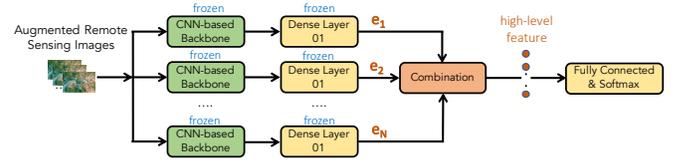


Fig. 2. The Teacher model generated from individual high-performance networks

layer to the global pooling layer, trained from the up-stream task using the ImageNet dataset [27], are transferred to the down-stream RSIC task. The Dense Layers block is considered to house the adapting layers for the down-stream RSIC task.

We also apply data augmentation for the RSIC down-stream task, namely Image Rotation [28] and Mixup [29], performed on the remote sensing image input dataset. In particular, all images in an original RSIC dataset are rotated using three different angles: 90, 180, and 270°. Since three angles are used, the augmented dataset is four times larger than the original. Next, batches of 60 images are randomly selected from the new dataset. For each batch, we apply the Mixup [29] method to mix the images within one batch with random ratios. Both Uniform and Beta distributions are used to generate the mixup ratios, and we make use of both the rotation augmented image database in addition to the new mixup images; as a consequence the batch size increases by three times from 60 to 180 images.

Thanks to the use of Mixup [29] for data augmentation, the labels will no longer be in one-hot encoding format. Therefore we apply Kullback-Leibler divergence (KL) loss [30] instead of Entropy loss to train the evaluating models, as in equation 1:

$$Loss_{KL}(\Theta) = \sum_{b=1}^B \sum_{c=1}^C y_{bc} \log \left\{ \frac{y_{bc}}{\hat{y}_{bc}} \right\} + \frac{\lambda}{2} \|\Theta\|_2^2, \quad (1)$$

where Θ presents trainable parameters, the constant λ is empirically set to 0.0001, the batch size B , the number of classes C , y_{bc} and \hat{y}_{bc} denote expected and predicted probabilities of an input image, respectively. Note that we set the low learning rate to be 0.0001 and none of trainable parameters are frozen during the training process.

B. Phase II: Develop the teacher and extract high-level features from the teacher

Given N high-performance models selected from Phase I, we then develop and train the teacher architecture during this phase. Again, we leverage parameter based transfer learning techniques to develop the teacher as shown in Figure 2. In particular, the first layer to the Dense Layer 01 of Dense Layers block from all N high-performance networks described in Phase I are reused and then combined to generate a composite high-level feature. If we consider N vectors $\mathbf{e}_n \in R^{512}$ as the output of the Dense Layer 01, the Combination block used to generate the composite high-level feature in Figure 2 by,

$$f(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N) = \sum_{n=1}^N \mathbf{e}_n \odot \mathbf{w}_i + \mathbf{b} \quad (2)$$

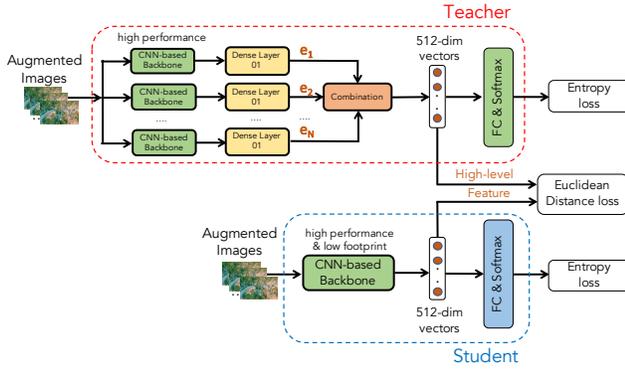


Fig. 3. The student model with knowledge distillation from the teacher.

where $\mathbf{w}_i, \mathbf{b} \in R^{512}$ are weight and bias trainable parameters. The high-level feature is finally transferred into a Fully Connected layer followed by a Softmax for classifying to target classes. When we finish training the teacher model, the high-level features are then extracted and used for the knowledge distillation process to train the student in Phase III which follows.

Data augmentation is used when training the teacher network, however only Image Rotation [28] is applied at this and thus the labels can remain in one-hot format, and Entropy loss can be used to train the teacher model as in equation 3:

$$Loss_E(\Theta) = - \sum_{b=1}^B \sum_{c=1}^C \mathbf{y}_{bc} \log \{\hat{\mathbf{y}}_{bc}\} + \frac{\lambda}{2} \|\Theta\|_2^2, \quad (3)$$

where Θ are trainable parameters, the constant λ is set to 0.0001, the batch size B and the number of classes C , \mathbf{y}_{bc} and $\hat{\mathbf{y}}_{bc}$ denote expected and predicted probabilities of a particular image, respectively.

We again set the low learning rate to 0.0001, and the trainable parameters of the first layer to the Dense Layer 01 are frozen when training the teacher. In other words, only trainable parameters in the Combination block and in the finally Fully Connected layer are updated during the training process.

C. Phase III: Train the student network to achieve high-performance and low-complexity RSIC

From the results in Phase I, a network architecture, which not only performs well but also presents a low footprint, is selected and considered as the base student model. We then train the student with the high-level features extracted from the teacher mentioned in Phase II. As Figure 3 shows, the student is trained with two loss functions. While the first Entropy loss is used for the classification task, the Euclidean Distance loss helps to ensure the high-level features of the student become closer to the high-level features extracted from the teacher, effectively guiding the feature discrimination ability of the student. The ratio between both losses is empirically set to 0.5/0.5.

Regarding the data augmentation used to train the student, only Image Rotation [28] is applied. We also set the low

learning rate of 0.0001 and no trainable parameters are frozen during the student training phase.

III. EXPERIMENTS AND RESULTS

A. Dataset

In this paper, the benchmark dataset of NWPU-RESISC45 [10] is used to evaluate all state of the art and proposed models. The dataset, which was collected from more than 100 different countries and regions around the world, consists of 31,500 remote sensing images separated into 45 scene classes. Each class comprises 700 RGB images with a resolution of $256 \times 256 \times 3$. To compare with state-of-the-art systems, we comply with the original settings mentioned in [10]. We then split the NWPU-RESISC45 dataset into Training and Testing sets with two different ratios: 10%-90% and 20%-80%, respectively.

B. Evaluation metric

As the Accuracy (Acc.%) has been used as the main metric to compare performance among the RSIC systems, we also apply the metric in this paper. Additionally, as we aim to achieve a low complexity model for the RSIC task, we compute the number of trainable parameters (M) to compare against state-of-the-art RSIC systems.

C. Experimental settings

We constructed our proposed deep learning networks with Tensorflow using the Adam method [42] for optimization. The training and evaluating processes are conducted on two Titan RTX 24GB GPUs. The results presented in this paper are all the average scores from 10 individual experimental runs.

D. Results and Discussions

As experimental results show in Table I, we can see that ConvNeXt, EfficientNet, DenseNet based models are competitive and outperform MobileNet, ResNet and Inception based models. Particularly, the best network architectures of ConvNeXtLarge and ConvNeXtTiny achieve 95.3% and 93.0% accuracy, respectively. Around 2% worse than ConvNeXtLarge, the performance of EfficientNetB7 and DenseNet201 on the NWPU-RESSIC45 task are 93.6% and 93.3%, respectively. Meanwhile, their smaller variants named DenseNet121 and EfficientNetB0 achieve over 92% accuracy.

Although ConvNeXt, EfficientNet and DenseNet based models perform well among the evaluating network architectures, these involve large footprints. In particular, the three best variants, namely ConvNeXtLarge, EfficientNetB7, and DensNet201 have some of the largest parameter set sizes of 196.6, 65.1, and 19.1 M, respectively. Among the ConvNeXt, EfficientNet and DenseNet variants, only EfficientNetB0 combines a relatively good accuracy of 92.3% with a low complexity footprint (4.7 M parameters). As a result, we select EfficientNetB0 as the foundation network for the student model required in Phase III. We also note that DenseNet201, EfficientNetB7 and ConvNeXtLarge perform better than 93% and their general architectures are dissimilar to each other. We

TABLE I
PERFORMANCE COMPARISON OF BENCHMARK CNN BASED NETWORK ARCHITECTURES ON THE NWPU-RESISC45 TASK WITH A TRAINING/TESTING SPLIT OF 20/80.

Network	MobileNetV2	MobileNet	ResNet50	Resnet151V2	InceptionV3	InceptionResNetV2
Accuracy (%)	88.0	90.8	91.8	92.4	86.9	90.5
Parameters (M)	2.9	3.7	24.6	59.2	22.8	55.1
Network	DenseNet121	DenseNet201	EfficientNetB0	EfficientNetB7	ConvNeXtTiny	ConvNeXtLarge
Accuracy (%)	92.0	93.3	92.3	93.6	93.0	95.3
Parameters (M)	7.5	19.1	4.7	65.1	27.5	196.6

TABLE II
PERFORMANCE COMPARISON OF THE TEACHER (A COMBINATION OF CONVNEXTLARGE, DENSENET201, EFFICIENTNETB7), THE STUDENT (EFFICIENTNETB0) WITH VARIOUS SETTINGS, ON THE NWPU-RESISC45 TASK WITH A TRAINING/TESTING SPLIT OF 20/80.

Network	Accuracy (%)	Parameters (M)
Teacher	96.2	280.8
EfficientNetB0 (student)	92.3	4.7
EfficientNetB0+distillation	94.8	4.7
EfficientNetB0-6B+distillation	94.4	3.0
EfficientNetB0-5B+distillation	93.5	0.93
EfficientNetB0-4B+distillation	91.3	0.37
EfficientNetB0-3B+distillation	85.6	0.11

TABLE III
PERFORMANCE (ACC.%) COMPARISON OF THE PROPOSED TEACHER AGAINST STATE-OF-THE-ART RSIC SYSTEMS ON THE NWPU-RESISC45 BENCHMARK WITH TWO SPLIT ARRANGEMENTS, AND WITHOUT ANY TRAINABLE PARAMETER SIZE CONSTRAINT.

Methods	10% training	20% training
MG-CAP [31]	90.8	93.0
EfficientNet-B3-aux [32]	91.1	93.8
ResNeXt-101 + MTL [33]	91.9	94.2
MBLANet [34]	92.3	94.6
GRMANet [35]	93.2	94.7
KFBNet [36]	93.1	95.1
CTNet [37]	93.9	95.4
TRS [22]	93.1	95.6
RSP-ViTAEv2-S-E100 [23]	94.4	95.6
Our system (Teacher)	94.6	96.2

therefore, select these three network architectures to generate the teacher, as required in Phase II.

As Table II shows, the teacher (i.e. a combination of DenseNet201, EfficientNetB7 and ConvNeXtLarge) achieves an accuracy of 96.2%, but with a very large footprint of 280.8 M parameters. Knowledge distillation from this capable teacher into student EfficientNetB0 allows it to achieve an accuracy of 94.8% while maintaining a low complexity of 4.7 M parameters. To propose a wide range of low complexity models, we further evaluate variants of the student EfficientNetB0 model. In particular, variants of the student are generated by removing certain convolutional blocks in the EfficientNetB0 backbone architecture to reduce complexity further. EfficientNetB0-6B to EfficientNetB0-3B are variants of EfficientNetB0 obtained by removing convolutional block 7 only, removing convolutional blocks 6 and 7, removing all convolutional blocks from 5 to 7 and removing all convolutional blocks 4 to 7 inclusive. Experimental results in Table II indicate that when the footprint of EfficientB0 based students is reduced, the accuracy performance also tends to decrease. However, we can achieve a very low complexity

TABLE IV
PERFORMANCE COMPARISON OF THE PROPOSED STUDENT AGAINST STATE-OF-THE-ART SYSTEMS ON THE BENCHMARK NWPU-RESISC45 DATASET WITH TWO SPLIT SETTINGS AND A CONSTRAINT OF NO MORE THAN 5M TRAINABLE PARAMETERS.

Methods	10% training	20% training
EfficientNet-B0-aux (\approx 5M) [32]	90.0	92.9
DMP-MobileNetV2 (3.47 M) [38]	90.3	93.1
BiMobileNet (2.52 M) [39]	91.9	93.9
SE-MDPMNet (5.17 M) [40]	91.8	94.1
LGRIN (4.63 M) [41]	91.9	94.4
Our system (Student+distillation)	93.3	94.8

model of 0.37 M parameters with a performance of 91.3% from EfficientNetB0-4B, which opens the potential for RSIC applications on a very wide range of edge devices.

Finally, we compare our proposed models to the state-of-the-art RSIC systems basing on two criteria: (1) accuracy performance without any model complexity constraint and (2) accuracy performance with a constraint of 5 M trainable parameters maximum. As Table III shows, RSIC performance with the first criterion reveals that our proposed teacher (i.e. a combination of ConvNeXtLarge, DenseNet201, and EfficientNetB7) outperforms the state-of-the-art systems, achieving 94.6% and 96.2% for the training/testing settings of 10/90 and 20/80, respectively. For the second criteria, i.e. low-complexity RSIC models (< 5 M trainable parameters) shown in Table IV, our proposed student with knowledge distillation also outperforms the state-of-the-art systems on both training/testing split arrangements, yielding results of 93.3% for a 10/90 split ratio and 94.8% for a 20/80 split ratio.

IV. CONCLUSION

This paper has presented, explored, and developed a range of deep convolutional neural networks for the remote sensing image classification (RSIC) task, and in particular considered model complexity. Through experimentation on the NWPU-RESISC45 benchmark, we obtained two RSIC systems: (1) a teacher developed by combining ConvNeXtLarge, DenseNet201, and EfficientNetB7 network architectures and; (2) a low complexity student (just 4.7 M trainable parameters), which leverages EfficientNetB0 via knowledge distillation from the teacher. Our proposed RSIC systems outperform the state of the art, whether complexity is constrained or not. Additionally, a wide range of low- to very low-complexity models using variants of EfficientNetB0 are proposed and explored, which are feasible to apply on edge devices with differing degrees of computational constraint.

REFERENCES

- [1] D. Poursanidis and N. Chrysoulakis, "Remote sensing, natural hazards and the contribution of ESA sentinel missions," *Remote Sensing Applications: Society and Environment*, vol. 6, pp. 25–38, 2017.
- [2] Q. Feng, J. Liu, and J. Gong, "UAV remote sensing for urban vegetation mapping using random forest and texture analysis," *Remote sensing*, vol. 7, no. 1, pp. 1074–1094, 2015.
- [3] C. J. Van Westen, "Remote sensing and GIS for natural hazards assessment and disaster risk management," *Treatise on geomorphology*, vol. 3, pp. 259–298, 2013.
- [4] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. SIGSPATIAL*, 2010, pp. 270–279.
- [5] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119–132, 2014.
- [6] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki, and R. Nemani, "Deepsat: a learning framework for satellite imagery," in *Proc. SIGSPATIAL*, 2015, pp. 1–10.
- [7] B. Zhao, Y. Zhong, G.-S. Xia, and L. Zhang, "Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 4, pp. 2108–2123, 2015.
- [8] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.
- [9] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1155–1167, 2018.
- [10] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.
- [11] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 6, pp. 703–715, 2001.
- [12] M. Pietikäinen, "Local binary patterns," *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.
- [13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886–893.
- [14] G. Lowe, "Sift-the scale invariant feature transform," *Int. J.*, vol. 2, no. 91–110, p. 2, 2004.
- [15] P. Du, J. Xia, W. Zhang, K. Tan, Y. Liu, and S. Liu, "Multiple classifier system for remote sensing image classification: A review," *Sensors*, vol. 12, no. 4, pp. 4764–4792, 2012.
- [16] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. ECCV*, 2010, pp. 143–156.
- [17] M. Mehmood, A. Shahzad, B. Zafar, A. Shabbir, and N. Ali, "Remote sensing image classification: A comprehensive review and applications," *Mathematical Problems in Engineering*, vol. 2022, 2022.
- [18] Y. Gu, Y. Wang, and Y. Li, "A survey on deep learning-driven remote sensing image scene understanding: Scene classification, scene retrieval and scene-guided object detection," *Applied Sciences*, vol. 9, no. 10, p. 2110, 2019.
- [19] A. Shabbir, N. Ali, J. Ahmed, B. Zafar, A. Rasheed, M. Sajid, A. Ahmed, and S. H. Dar, "Satellite and scene image classification based on transfer learning and fine tuning of resnet50," *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [20] W. Tong, W. Chen, W. Han, X. Li, and L. Wang, "Channel-attention-based densenet network for remote sensing image scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4121–4132, 2020.
- [21] D. Zhang, Z. Liu, and X. Shi, "Transfer learning on efficientnet for remote sensing image classification," in *ICMCCE*, 2020, pp. 2255–2258.
- [22] J. Zhang, H. Zhao, and J. Li, "Trs: Transformers for remote sensing scene classification," *Remote Sensing*, vol. 13, no. 20, p. 4143, 2021.
- [23] D. Wang, J. Zhang, B. Du, G.-S. Xia, and D. Tao, "An empirical study of remote sensing pretraining," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [24] Z. Sun, R. Sun, L. Lu, and A. Mislove, "Mind your weight (s): A large-scale study on insufficient machine learning model protection in mobile apps," in *Proc. USENIX*, 2021, pp. 1955–1972.
- [25] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.
- [26] F. Chollet *et al.*, "Keras," <https://keras.io>, 2015.
- [27] O. R. *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, no. 3, pp. 211–252, 2015.
- [28] C. Shorten and T. M. Khoshgofaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [29] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.
- [30] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [31] S. Wang, Y. Guan, and L. Shao, "Multi-granularity canonical appearance pooling for remote sensing scene classification," *IEEE Transactions on Image Processing*, vol. 29, pp. 5396–5407, 2020.
- [32] Y. Bazi, M. M. Al Rahhal, H. Alhichri, and N. Alajlan, "Simple yet effective fine-tuning of deep CNNs using an auxiliary classification loss for remote sensing scene classification," *Remote Sensing*, vol. 11, no. 24, p. 2908, 2019.
- [33] Z. Zhao, Z. Luo, J. Li, C. Chen, and Y. Piao, "When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework," *Remote Sensing*, vol. 12, no. 20, p. 3276, 2020.
- [34] S.-B. Chen, Q.-S. Wei, W.-Z. Wang, J. Tang, B. Luo, and Z.-Y. Wang, "Remote sensing scene classification via multi-branch local attention network," *IEEE Transactions on Image Processing*, vol. 31, pp. 99–109, 2021.
- [35] B. Li, Y. Guo, J. Yang, L. Wang, Y. Wang, and W. An, "Gated recurrent multiattention network for VHR remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2021.
- [36] F. Li, R. Feng, W. Han, and L. Wang, "High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 11, pp. 8077–8092, 2020.
- [37] P. Deng, K. Xu, and H. Huang, "When CNNs meet vision transformer: A joint framework for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.
- [38] J. Hou, Z. Guo, Y. Wu, W. Diao, and Y. Feng, "Dmpconv: Decoupling multi-branch pointwise convolutions for light-weight remote sensing scene classification," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, no. 3, 2022.
- [39] D. Yu, Q. Xu, H. Guo, C. Zhao, Y. Lin, and D. Li, "An efficient and lightweight convolutional neural network for remote sensing image scene classification," *Sensors*, vol. 20, no. 7, p. 1999, 2020.
- [40] B. Zhang, Y. Zhang, and S. Wang, "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*, vol. 12, no. 8, pp. 2636–2653, 2019.
- [41] C. Xu, G. Zhu, and J. Shu, "A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2021.
- [42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.