Enhanced Aerial Scene Classification Through ConvNeXt Architectures and Channel Attention

Leo Thomas Ramos and Angel D. Sappa

Abstract This work explores the integration of a Channel Attention (CA) module into the ConvNeXt architecture to improve performance in scene classification tasks. Using the UC Merced dataset, experiments were conducted with two data splits: 50% and 20% for training. Models were trained for up to 20 epochs, limiting the training process to assess which models could extract the most relevant features efficiently under constrained conditions. The ConvNeXt architecture was modified by incorporating a Squeeze-and-Excitation block, aiming to enhance the importance of each feature channel. ConvNeXt models with CA showed strong results, achieving the highest performance in the experiments conducted. ConvNeXt large with CA reached 90% accuracy and 89.75% F1-score with 50% of the training data, while ConvNeXt base with CA achieved 77.14% accuracy and 75.23% F1-score when trained with only 20% of the data. These models consistently outperformed their standard counterparts, as well as other architectures like ResNet and Swin Transformer, achieving improvements of up to 9.60% in accuracy, highlighting the effectiveness of CA in boosting performance, particularly in scenarios with limited data.

Keywords Scene classification · Remote sensing · Convolutional neural networks · Attention mechanisms · Computer vision

L. T. Ramos · A. D. Sappa
Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain
e-mail: asappa@cvc.uab.cat

L. T. Ramos (✉)
Kauel Inc., Menlo Park, USA
e-mail: ltramos@cvc.uab.cat

A. D. Sappa
ESPOL Polytechnic University, Guayaquil, Ecuador

X.-S. Yang et al. (eds.), *Proceedings of Tenth International Congress on Information and Communication Technology*, Lecture Notes in Networks and Systems 1412,
https://doi.org/10.1007/978-981-96-6429-0_19

1 Introduction

Scene Classification (SC) is a computer vision task that categorizes image or video frames into predefined categories [2, 12]. Unlike object classification, which identifies individual foreground objects, SC analyzes the entire image content to determine its semantic meaning [2]. This involves examining elements like color, texture, shape, spatial object distribution, and their semantic relationships [3, 7], enabling categorization based on the scene's holistic composition. For this reason, SC is applicable across various levels and contexts, with the aerial or remote sensing standing out as a prominent research focus [11, 13]. Its primary objective is to label land cover types (e.g., beach, forest, desert) or land use categories (e.g., agricultural area, baseball field, airport) [7, 11]. This approach is particularly valuable for applications such as urban planning, resource management, environmental monitoring, and disaster response decision-making [13].

Traditional SC methods relied on handcrafted features [3], which had limited representational capacity [2], particularly for low and mid-level features lacking high-level semantic information [3]. The advent of Deep Learning (DL), especially Convolutional Neural Networks (CNNs) [8], has progressively replaced handcrafted features. More recently, Transformers have further improved scene classification by capturing global dependencies in data, solidifying DL as the dominant paradigm for this task [7]. Nevertheless, SC remains a challenging task, particularly in aerial perspectives, due to diverse objects, complex backgrounds, and intricate spatial patterns [1, 7]. Accurate classification is further hindered by the need for large amounts of labeled data for model training [8], which is often impractical and resource-intensive, especially for advanced models like Transformers. Thus, enhancing data efficiency and model accuracy remains a central research focus.

CNN-based approaches are efficient up to a certain depth, but their computational cost rises significantly as layers increase, often without proportional performance gains [5]. In contrast, Transformers leverage Attention Mechanisms (AMs) to achieve superior performance in various computer vision tasks, frequently surpassing CNNs. However, their high computational demands [14] present a major drawback in scenarios with limited resources and data. On this basis, this work explores the use of the ConvNeXt architecture [6] for scene classification. ConvNeXt, a CNN model inspired by Transformer design principles, aims to deliver high performance while retaining the computational efficiency of CNNs. To enhance its performance, we introduced a Squeeze-and-Excitation block as an AM between the feature extractor and classifier. This recalibrates channel-wise feature responses, enabling the model to focus on the most relevant features, thereby improving feature representation and classification accuracy.

To evaluate the proposed ConvNeXt-based approach, we used the UC Merced dataset, comprising 21 classes with 100 images each. Two configurations were tested: using 50% and 20% of the data for training, both under 20 epochs. This setup aims to demonstrate that the proposed approach is capable of extracting rich features and achieving accurate classification while requiring fewer computational resources during training. Comparisons with ResNet and Transformers show that the ConvNeXt-based approach outperforms them while maintaining lower training and inference times. Building on these results, this work aims to contribute to the field of scene classification by balancing accuracy and efficiency, making it suitable for deployment in resource-limited scenarios.

2 Materials and Methods

2.1 Dataset Description

The dataset used for this research is the UC Merced dataset, developed by researchers at the University of California, Merced, USA. It comprises 21 land use classes, with 100 RGB images for each class. Each image is in TIFF format, with a size of 256×256 pixels and a spatial resolution of 0.3 meters per pixel. The images were manually extracted from larger images in the USGS National Map Urban Area Imagery collection, covering various regions across the United States. This dataset is widely used in remote sensing and scene classification tasks, serving as a benchmark for evaluating model performance. Examples of the different classes in this dataset can be seen in Fig. 1. The dataset is available for free and can be accessed through the authors' website.[1]

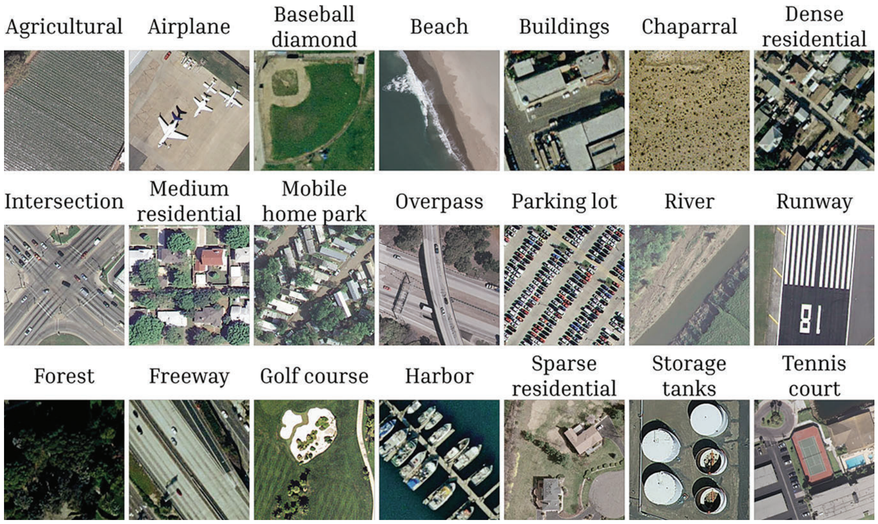

Fig. 1 UC Merced dataset and representative samples from each class

[1] http://weegee.vision.ucmerced.edu/datasets/landuse.html.

2.2 Model Description

ConvNeXt Overview The ConvNeXt architecture was developed by researchers at Facebook with the goal of modernizing CNNs. They based their work on a ResNet50 architecture and implemented several design modifications inspired by the Vision Transformer [6]. Key modifications include the use of a multi-stage design, where each stage has a different feature map resolution, similar to the Swin Transformer; the incorporation of the inverted bottleneck from the Vision Transformer, which reduces FLOPs and improves performance; increasing the kernel sizes from 3×3 to 7×7 for enhanced performance without a significant increase in FLOPs; replacing the ReLU activation function with GELU; reducing the number of normalization layers; and replacing batch normalization with layer normalization. These changes enabled ConvNeXt to achieve remarkable performance, even surpassing Swin Transformers in some cases. Figure 2 shows a comparison between the ConvNeXt block and the models that inspired its design. ConvNeXt comes in four versions: tiny, small, base, and large, each offering different levels of depth.

Channel Attention Module AMs are modules that enhance neural networks by prioritizing relevant input features [9], improving their ability to capture meaningful patterns. These mechanisms operate across dimensions like spatial, temporal, or

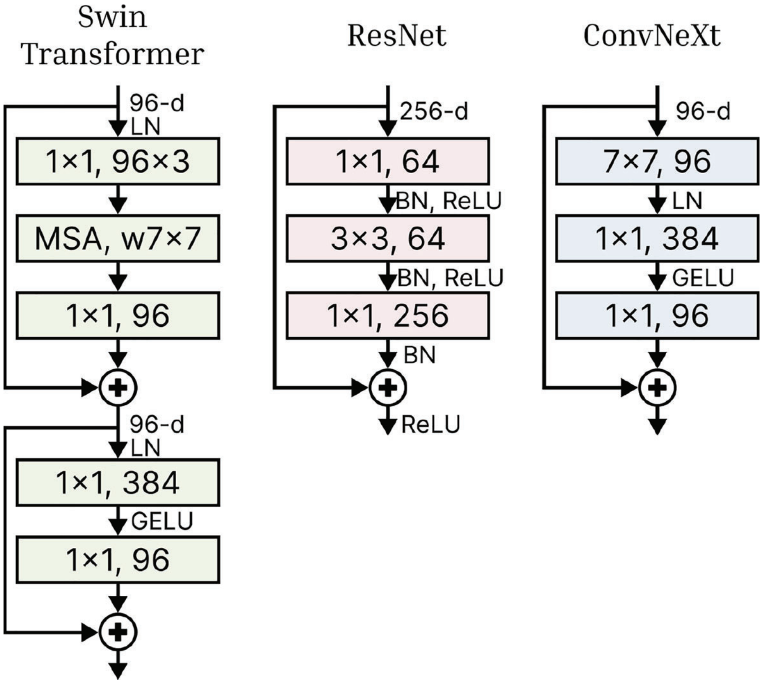

Fig. 2 Comparison between Swin Transformer, ResNet, and ConvNeXt blocks

channel. This research utilizes CA, which determines the importance of each feature channel [10], enabling models to amplify informative channels while suppressing less relevant ones, thereby boosting performance.

While self-attention mechanisms in Transformers are highly effective, we chose to integrate CA into ConvNeXt for its computational efficiency in enhancing feature representations without adding significant complexity. This approach balances accuracy and efficiency, essential for our application. The CA module is based on the Squeeze-and-Excitation (SE) block [4], which recalibrates feature channels by aggregating global context. This design was chosen for its ability to enhance feature discrimination while maintaining low computational cost.

The CA module operates in two phases: squeeze and excitation. In the squeeze phase, global average pooling reduces the spatial dimensions of the input feature map, producing a vector that captures the global context of each channel. In the excitation phase, the vector passes through two fully connected layers: the first reduces dimensionality using a reduction factor, and the second restores it. A ReLU activation introduces non-linearity between the layers, and a sigmoid function normalizes the recalibrated weights to a range of 0 to 1. Finally, the input feature map is scaled by multiplying each channel with its corresponding attention weight, recalibrating channels based on global importance. Figure 3 illustrates the CA module.

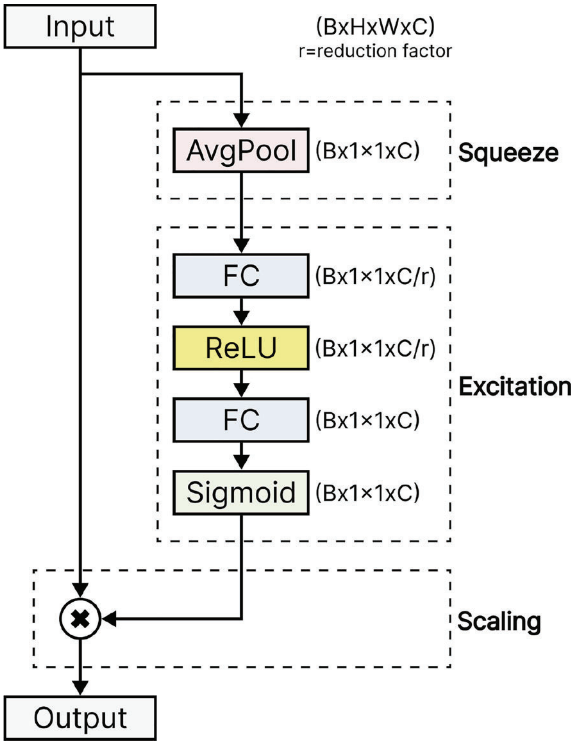

Fig. 3 Channel attention module used in this work

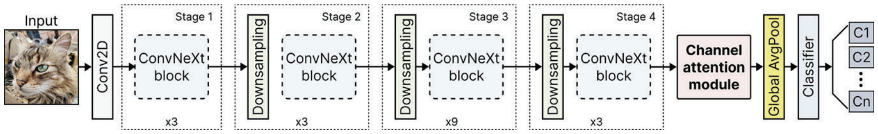

Fig. 4 ConvNeXt architecture (tiny version) with channel attention module

ConvNeXt with Channel Attention The ConvNeXt architecture comprises four stages, each containing an increasing number of ConvNeXt blocks to process the input data. As the input progresses, spatial resolutions decrease, reducing feature map sizes while increasing depth, enabling the network to learn a hierarchy of features-from simple patterns in early stages to complex representations in later stages. Downsampling between stages uses convolutional operations with a stride of 2 for efficient processing. These stages form the core of feature extraction. Afterward, adaptive average pooling reduces spatial dimensions to a single value per channel, summarizing the feature map. The classifier normalizes these features with layer normalization, flattens them into a 1D vector, and maps this vector to output classes via a fully connected layer, producing the network's final predictions.

To integrate the CA module into the ConvNeXt architecture, it is placed after the four feature extraction stages, as illustrated in Fig. 4. This placement ensures that, after learning progressively abstract features, the CA mechanism recalibrates the importance of each feature channel before pooling. By applying attention while spatial dimensions are intact, the module utilizes full spatial information to assign appropriate weights to each channel. After recalibration, adaptive average pooling reduces spatial dimensions, and the features are passed to the classifier for final predictions. This integration enhances feature representation, improving classification accuracy while maintaining model efficiency.

2.3 Implementation Details

The implementation was performed in Python using the PyTorch framework, with ConvNeXt sourced from PyTorch's vision model library.[2] The dataset was split into two configurations: 50% (1,050 images) for training and 50% for testing in the first split, and 20% (420 images) for training and 80% for testing in the second. To ensure consistency, the 50% testing set from the first split was excluded from the 20% training set in the second split, preventing overlap and enabling evaluation on a shared testing set. Images were resized to 224×224 and normalized using ImageNet's mean and standard deviation.

The training was limited to 20 epochs, saving the best model based on the lowest loss. This constraint aimed to evaluate the models' ability to quickly extract relevant

[2] https://pytorch.org/vision/stable/models.html.

features and perform well under a short training regime. The training parameters were set uniformly across all models, with Adam as the optimizer, a learning rate of 1×10^{-4}, cross-entropy as the loss function, and a batch size of 64. The hardware used consisted of two Nvidia A100 SXM4 40 GB GPUs, 64 CPU cores, and 128 GB of RAM. The evaluation of the models was carried out using the accuracy (Equation 1) and F1-score (Equation 2) metrics.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

$$\text{F1-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{2}$$

3 Results and Discussion

Table 1 presents the results for models trained on 50% of the data, highlighting the superior performance of ConvNeXt architectures with CA. ConvNeXt Large achieves the highest accuracy (90%) and F1-score (89.75%) among all models. The base, small, and tiny versions also outperform ConvNeXt without CA, as well as ResNet and Swin Transformer. For instance, ConvNeXt Tiny with CA achieves 88.19% accuracy, showing improvements of 3.23% over ResNet50, 4.99% over Swin Transformer Tiny, and 1.09% over ConvNeXt Tiny without CA.

Table 1 Evaluation results of models with 50% training data on testing set

Method	Accuracy	F1-score	Training time (seconds)	Inference time (seconds)
ResNet50	0.8543	0.8507	**41.07**	**0.0005**
ResNet101	0.8705	0.8676	54.27	0.0010
ResNet152	0.8600	0.8563	68.46	0.0018
Swin Transformer tiny	0.8400	0.8351	51.70	0.0007
Swin Transformer small	0.8771	0.8736	77.29	0.0012
Swin Transformer base	0.8752	0.8701	96.89	0.0010
ConvNeXt tiny	0.8724	0.8676	50.86	**0.0005**
ConvNeXt small	0.8810	0.8757	75.35	0.0008
ConvNeXt base	0.8857	0.8806	98.78	0.0008
ConvNeXt large	0.8876	0.8842	163.28	0.0009
ConvNeXt tiny CA	0.8819	0.8785	52.65	0.0009
ConvNeXt small CA	0.8905	0.8891	75.61	0.0008
ConvNeXt base CA	0.8971	0.8930	102.59	0.0010
ConvNeXt large CA	**0.9000**	**0.8975**	164.91	0.0009

Similarly, ConvNeXt Base with CA achieves a 1.29% accuracy improvement over ConvNeXt Base without CA, 2.50% over Swin Transformer Base, and 4.31% over ResNet152. In terms of F1-score, it shows improvements of 1.41%, 2.63%, and 4.29%, respectively. These results highlight the effectiveness of CA in enhancing performance, even under limited training conditions, by improving feature extraction and generalization.

Regarding training times, ResNet50 is the fastest model, completing training in 41.07 s, while ConvNeXt large with CA is the slowest at 164.91 s. However, despite the apparent large difference, it is important to note that 164.91 s is just over 2.7 min, which is still very efficient. Similarly, the inference times do not show excessive variation, as ResNet50, one of the fastest in inference, reports 0.0005 s, just milliseconds faster than ConvNeXt large with CA, which takes 0.0009 s. The remaining models, both in training and inference times, report figures within similar ranges.

For a detailed analysis, Fig. 5 presents confusion matrices of the best models from each architecture family listed in Table 1. ConvNeXt Large with CA shows notable improvements in specific classes, such as river, compared to its non-attention counterpart and Swin Transformer Small. In the mobile home park class, it achieves a 94% correct classification rate, outperforming ConvNeXt Large without CA (86%), ResNet101 (84%), and Swin Transformer Small (86%). It also demonstrates better performance in classes like airplane and runway, though with smaller margins. These results confirm that CA integration enhances performance across multiple classes.

Table 2 presents results for models trained on 20% of the data. As expected, performance decreases due to the limited data, constraining the models' learning capacity. Despite this, ConvNeXt models with CA again outperform other architectures. ConvNeXt Base with CA achieves the highest performance, with 77.14% accuracy and 75.23% F1-score. Notably, robust models (e.g., ResNet152, Swin Transformer Base, ConvNeXt Large) underperform compared to their lighter variants, likely due to difficulties fitting their larger parameter counts in this limited data regime. However, ConvNeXt Large with CA still shows significant gains over robust counterparts, with a 2.10% accuracy increase over its non-attention version, 9.60% over Swin Transformer Base, and 3.81% over ResNet152. These results underscore the effectiveness of CA in enabling ConvNeXt to perform well, even under highly constrained conditions where other robust models falter.

Moving forward, the training times decrease overall compared to the previous experiment, which is expected given the reduction in the number of training images. The times range from 29.65 s for ResNet50 to 104.30 s for ConvNeXt large with CA. Once again, all training times fall within an optimal and efficient range. Inference times remain almost constant, with slight differences of 1 or 2 ms compared to the previous experiment in a few cases.

Figure 6 shows the confusion matrices for the best models trained on 20% of the data. Consistent with Table 2, the matrices highlight weaknesses across all models. However, ConvNeXt Base with CA excels by achieving 100% accuracy in six

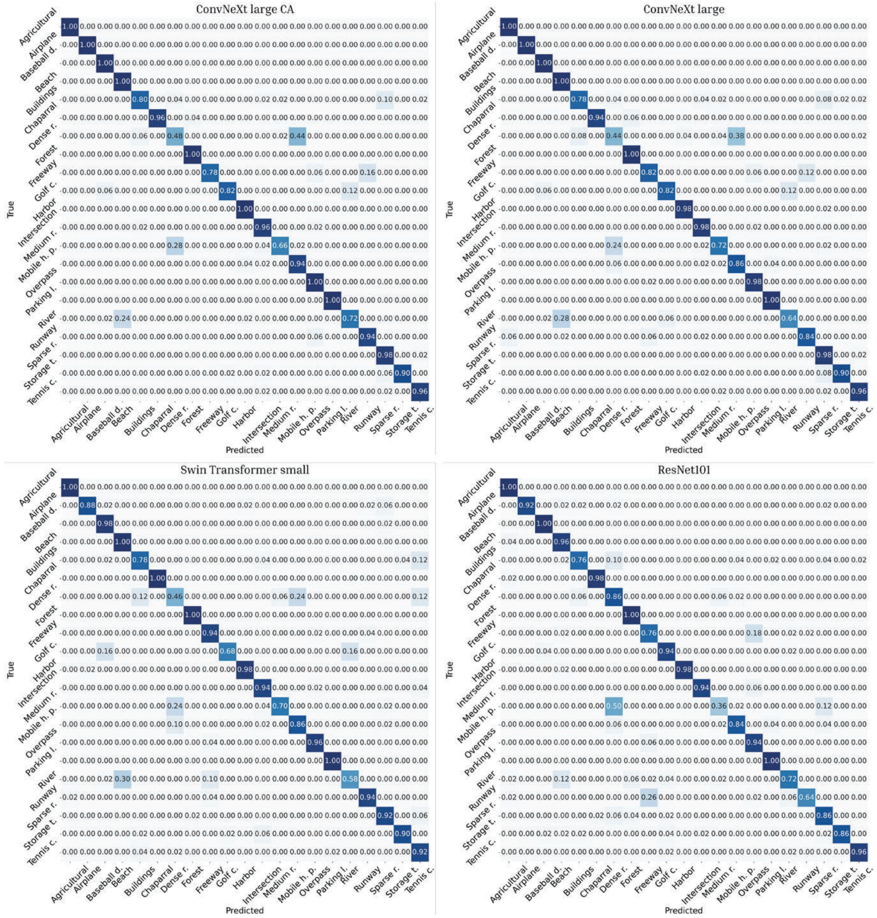

Fig. 5 Confusion matrices of the best models with 50% training data

classes: agricultural, baseball diamond, chaparral, forest, mobile home park, and overpass, outperforming other models with fewer perfect predictions. It also shows slight improvements in classes like sparse residential and runway. Although it underperforms in some classes, such as medium residential and tennis court, the performance gap remains small. Combined with its strengths, these results confirm its overall superiority over other approaches.

Table 2 Evaluation results of models with 20% training data on testing set

Method	Accuracy	F1-score	Training time (seconds)	Inference time (seconds)
ResNet50	0.7200	0.6970	**29.65**	0.0006
ResNet101	0.7238	0.7049	37.65	0.0010
ResNet152	0.7119	0.6806	47.01	0.0018
Swin Transformer tiny	0.6752	0.6485	35.45	0.0006
Swin Transformer small	0.6781	0.6596	52.84	0.0011
Swin Transformer base	0.6743	0.6458	62.40	0.0011
ConvNeXt tiny	0.7219	0.6992	34.67	**0.0005**
ConvNeXt small	0.7419	0.7175	47.45	0.0008
ConvNeXt base	0.7686	0.7416	62.86	0.0008
ConvNeXt large	0.7238	0.7014	103.27	0.0008
ConvNeXt tiny CA	0.7543	0.7368	34.94	**0.0005**
ConvNeXt small CA	0.7457	0.7306	49.05	0.0008
ConvNeXt base CA	**0.7714**	**0.7523**	63.09	0.0008
ConvNeXt large CA	0.7390	0.7131	104.30	0.0008

4 Conclusions and Future Works

This work studies the integration of a CA module into the ConvNeXt architecture to enhance its performance in scene classification tasks. The modification recalibrates feature channel importance, improving the model’s ability to extract relevant information efficiently. Experiments were conducted on the UC Merced dataset under two training regimes: 50% and 20% of the data, with a 20-epoch limit designed to test the models in a constrained training scenario, where only the most capable models, able to extract relevant features efficiently, would perform well. The results show that ConvNeXt models with CA consistently outperform their counterparts without attention, as well as other architectures like ResNet and Swin Transformer. ConvNeXt large with CA achieved the highest accuracy and F1-score when trained with 50% of the data, reaching 90% accuracy and 89.75% F1-score. Even under more limited data conditions, such as training with only 20% of the data, ConvNeXt base with CA demonstrates strong performance, achieving 77.14% accuracy and 75.23% F1-score. These findings highlight the significant impact of CA in enhancing ConvNeXt, especially in data-constrained scenarios. This work demonstrates the potential of integrating ConvNeXt with CA for improved scene classification in resource- and data-limited scenarios. Future studies could explore alternative attention mechanisms, apply this approach to other datasets, and compare it with additional architectures to better understand its strengths and limitations across diverse tasks.

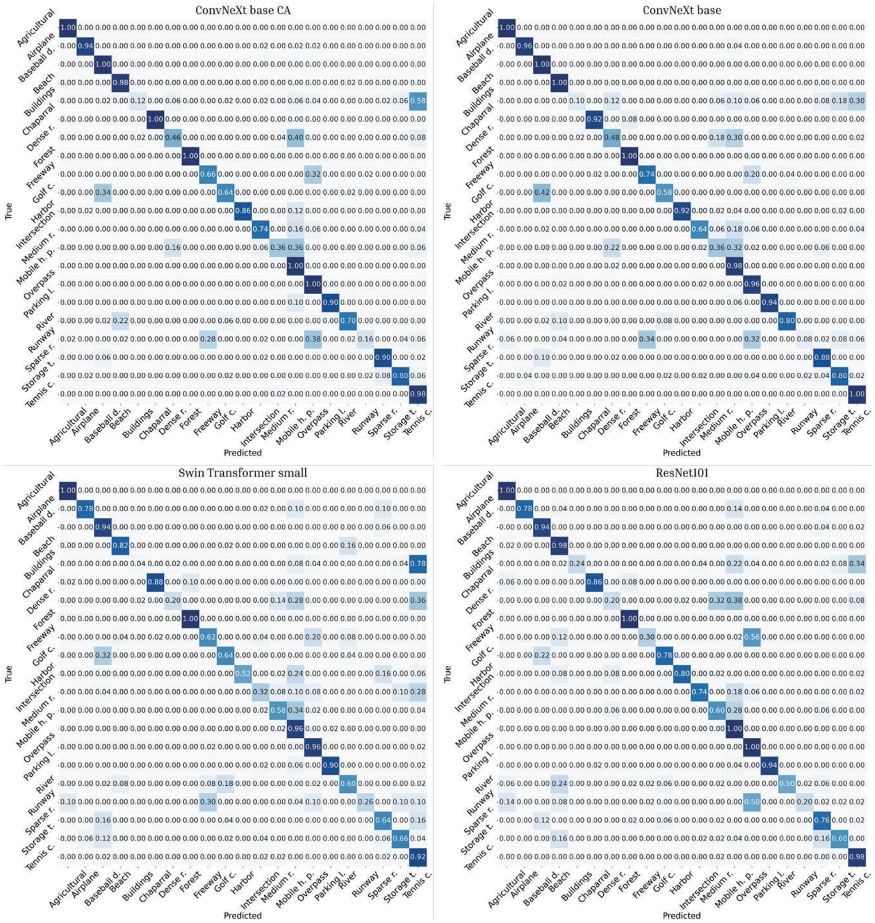

Fig. 6 Confusion matrices of the best models with 20% training data

Acknowledgements This was supported in part by Grant PID2021-128945NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”. The authors acknowledge the support of the Generalitat de Catalunya CERCA Program to CVC’s general activities, and the Departament de Recerca i Universitats from Generalitat de Catalunya with reference 2021SGR 01499.

References

1. Chen SB, Wei QS, Wang WZ, Tang J, Luo B, Wang ZY (2022) Remote sensing scene classification via multi-branch local attention network. IEEE Trans Image Process 31:99–109. https://doi.org/10.1109/TIP.2021.3127851
2. Cheng G, Xie X, Han J, Guo L, Xia GS (2020) Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. IEEE J Sel Top Appl Earth Obs Rem Sens 13:3735–3756. https://doi.org/10.1109/JSTARS.2020.3005403
3. Guo N, Jiang M, Wang D, Zhou X, Song Z, Li Y, Gao L, Luo J (2024) Scene classification for remote sensing image of land use and land cover using dual-model architecture with multilevel feature fusion. Int J Digit Earth 17(1). https://doi.org/10.1080/17538947.2024.2353166
4. Hu J, Shen L, Albanie S, Sun G, Wu E (2020) Squeeze-and-excitation networks. IEEE Trans Pattern Anal Mach Intell 42(8):2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372
5. Kim B, Yuvaraj N, Sri Preethaa KR, Arun Pandian R (2021) Surface crack detection using deep learning with shallow cnn architecture for enhanced computation. Neural Comput Appl 33(15):9289–9305. https://doi.org/10.1007/s00521-021-05690-8
6. Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11966–11976. https://doi.org/10.1109/CVPR52688.2022.01167
7. Ma A, Wan Y, Zhong Y, Wang J, Zhang L (2021) Scenenet: Remote sensing scene classification deep learning network using multi-objective neural evolution architecture search. ISPRS J Photogram Rem Sens 172:171–188. https://doi.org/10.1016/j.isprsjprs.2020.11.025
8. Mei S, Yan K, Ma M, Chen X, Zhang S, Du Q (2021) Remote sensing scene classification using sparse representation-based framework with deep feature fusion. IEEE J Select Top Appl Earth Obs Rem Sens 14:5867–5878. https://doi.org/10.1109/JSTARS.2021.3084441
9. Niu Z, Zhong G, Yu H (2021) A review on the attention mechanism of deep learning. Neurocomputing 452:48–62. https://doi.org/10.1016/j.neucom.2021.03.091
10. Ramos LT, Sappa AD (2024) Multispectral semantic segmentation for land cover classification: An overview. IEEE J Select Top Appl Earth Obs Rem Sens 17:14295–14336. https://doi.org/10.1109/JSTARS.2024.3438620
11. Tang X, Ma Q, Zhang X, Liu F, Ma J, Jiao L (2021) Attention consistent network for remote sensing scene classification. IEEE J Select Top Appl Earth Obs Rem Sens 14:2030–2045. https://doi.org/10.1109/JSTARS.2021.3051569
12. Tong W, Chen W, Han W, Li X, Wang L (2020) Channel-attention-based densenet network for remote sensing image scene classification. IEEE J Select Top Appl Earth Obs Rem Sens 13:4121–4132. https://doi.org/10.1109/JSTARS.2020.3009352
13. Wang J, Li W, Zhang M, Tao R, Chanussot J (2023) Remote-sensing scene classification via multistage self-guided separation network. IEEE Trans Geosci Rem Sens 61:1–12. https://doi.org/10.1109/TGRS.2023.3295797
14. Zhu C, Ping W, Xiao C, Shoeybi M, Goldstein T, Anandkumar A, Catanzaro B (2021) Long-short transformer: Efficient transformers for language and vision. In: Advances in neural information processing systems, vol 34. Curran Associates, Inc., pp 17723–17736