# Comparative Analysis of CNN and Transformer Architectures for Remote Sensing Scene Classification

Akhiyar Waladi Universitas Jambi
Jambi, Indonesia
akhiyar.waladi@unja.ac.id

***Abstract*—How much does network architecture actually matter for remote sensing scene classification? We investigated this by benchmarking eight deep learning models on two standard datasets: EuroSAT (10 classes, 27,000 Sentinel-2 images) and UC Merced (21 classes, 2,100 aerial images). The models span three design families: classical CNNs (ResNet-50, ResNet-101, DenseNet-121, EfficientNet-B0, EfficientNet-B3), vision transformers (ViT-B/16, Swin Transformer), and a modernized CNN (ConvNeXt-Tiny). Every model was trained with the same hyperparameters, the same augmentation pipeline, and the same ImageNet-pretrained initialization. ConvNeXt-Tiny reached the highest accuracy on EuroSAT (99.06%) and EfficientNet-B3 on UC Merced (99.76%), but the gap between the best and worst model was less than one percentage point on both datasets. McNemar's test showed that most pairwise differences were not statistically significant. EfficientNet-B0, the smallest model at 4.0M parameters, reached 98.54% and 99.52%, which raises the question of whether these benchmarks can still meaningfully separate architectures. We argue that for standard scene classification tasks with transfer learning, the training recipe matters more than the specific architecture.**

***Index Terms*—Scene classification, remote sensing, deep learning, convolutional neural networks, vision transformers, transfer learning, EuroSAT, UC Merced**

## I. Introduction

ASSIGNING a single land-use label to a satellite or aerial image patch is one of the oldest problems in remote sensing, and one that has seen large accuracy gains since deep learning entered the field [1], [2]. The task matters because automated scene classification feeds into urban expansion tracking, environmental monitoring, disaster mapping, and national land cover inventories.

Before deep learning, the standard pipeline relied on handcrafted features: color histograms, texture descriptors, bag-of-visual-words representations [3]. A researcher would design features by hand, feed them to a classifier like an SVM, and hope the chosen features captured the right information. This worked to a degree, but it was labor-intensive and did not scale well to large numbers of classes [4].

CNNs changed the game. Networks like VGGNet [5], ResNet [6], and DenseNet [7] learn their own features directly from pixels, and when pretrained on ImageNet [8] and fine-tuned on remote sensing data, they consistently beat handcrafted approaches [9], [10]. Transfer learning proved especially useful because labeled satellite imagery is often scarce [11], [12].

More recently, transformers have arrived in computer vision. The idea, first proposed for language modeling [13], is to process an image as a sequence of patches and let self-attention learn which patches relate to which. Vision Transformers (ViTs) [14] and their hierarchical variants like the Swin Transformer [15] have matched or beaten CNNs on general benchmarks, and researchers quickly began testing them on remote sensing data [16], [17]. At the same time, ConvNeXt [18] showed that a purely convolutional network, when trained with modern recipes borrowed from transformers, can reach the same accuracy level.

This creates an uncomfortable situation for practitioners. There are now three competing families of architectures (classical CNNs, vision transformers, modernized CNNs), each with papers claiming superiority. But most published comparisons test only two or three models, or use different training protocols, or evaluate on a single dataset. It is hard to know whether reported differences come from the architecture itself or from differences in hyperparameters, augmentation, or training schedule.

We set out to remove these confounding factors. We took eight architectures from all three families, gave each one the same ImageNet-pretrained weights, applied the same augmentation and optimization, and measured accuracy on two datasets that cover different imaging modalities and class counts. We then applied McNemar's test [19] to check whether any observed accuracy gaps were statistically real. We also recorded parameter counts and training times to assess efficiency.

## II. Related Work

### A. CNN-Based Scene Classification

ResNet [6] introduced shortcut connections that let gradients flow directly through the network, and the 50- and 101-layer variants quickly became the default baselines in remote sensing. DenseNet [7] took a different route: every layer receives input from all preceding layers, which encourages feature reuse and keeps the parameter count low. The EfficientNet family [20] showed that scaling depth, width, and resolution together is more effective than scaling any one dimension alone. Nogueira et al. [9] provided early evidence that fine-tuning ImageNet-pretrained CNNs beats training from scratch for aerial scene recognition, a finding that has been replicated many times since [2], [21].

### B. Transformer-Based Approaches

ViT [14] splits an image into fixed-size patches, embeds them, and feeds the sequence to a standard transformer encoder with self-attention. The appeal for remote sensing is that attention can capture relationships between distant image regions that local convolution filters would miss [17]. The main drawback is computational cost: self-attention scales quadratically with the number of patches. Swin Transformer [15] addresses this by computing attention inside local windows that shift across layers, producing a hierarchical feature pyramid at linear cost. Hong et al. [16] applied a transformer design specifically to hyperspectral data, showing that the architecture can also handle non-RGB inputs.

### C. Modernized CNNs

ConvNeXt [18] is the result of a thought experiment: what happens if you take a plain ResNet and, one design choice at a time, adopt ideas from transformers? Larger kernels ($7\times7$), layer normalization, GELU activations, and an inverted bottleneck layout brought a standard convolution network to the same accuracy as Swin Transformer on ImageNet. This raises an interesting question for remote sensing: if architecture matters less than training recipe on ImageNet, does the same hold for satellite and aerial imagery?

### D. Benchmark Datasets

The UC Merced Land Use dataset [3] has 2,100 aerial images across 21 land-use classes at 0.3 m resolution, drawn from USGS National Map imagery. EuroSAT [22] provides 27,000 Sentinel-2 multispectral patches at 10 m resolution with 10 classes. Larger collections exist, including NWPU-RESISC45 [4] (31,500 images, 45 classes) and AID [23] (10,000 images, 30 classes), but EuroSAT and UC Merced remain popular because they are freely available and small enough to run full experiments quickly. We chose these two specifically because they differ in resolution, image source (satellite vs. aerial), and number of classes, giving us two complementary testbeds.

## III. Methodology

Fig. 1 provides an overview of the research methodology. The pipeline consists of five phases: data acquisition, preprocessing, model training, performance evaluation, and comparative analysis.

### A. Datasets

*1) EuroSAT:* We use EuroSAT [22], a collection of 27,000 Sentinel-2 satellite patches in 10 land-use categories. Each patch is $64\times64$ pixels at 10 m ground sampling distance. The classes range from natural covers (Forest, SeaLake, River) to agricultural types (AnnualCrop, PermanentCrop, Pasture) and built-up areas (Highway, Industrial, Residential). We use the RGB bands only, since all eight architectures expect three-channel input. Fig. 2 shows one sample per class.

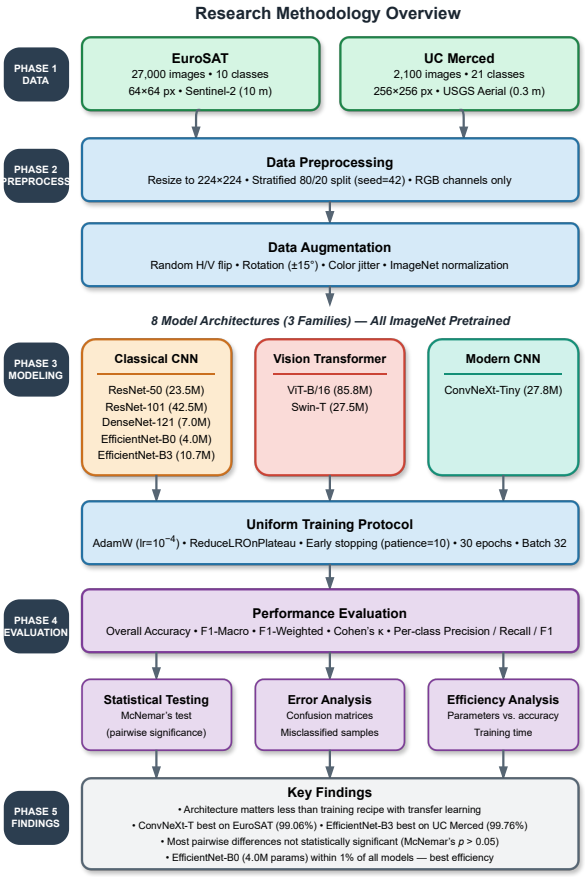


Fig. 1. Overview of the research methodology. Two benchmark datasets are preprocessed with a uniform pipeline and used to train eight architectures from three design families under identical conditions. The trained models are evaluated using classification metrics, statistical significance tests, error analysis, and computational efficiency measures.

*2) UC Merced Land Use:* The UC Merced dataset [3] has 2,100 aerial images at 0.3 m resolution, split evenly across 21 classes (100 images each, $256\times256$ pixels). The fine resolution means individual buildings, tennis courts, and storage tanks are clearly visible, but it also means that classes like denseresidential, mediumresidential, and sparseresidential differ only in the spacing between structures, which makes them easy to confuse. Samples from all 21 classes appear in Fig. 3.

*3) Data Partitioning:* We applied an 80/20 stratified random split with a fixed seed (42) for both datasets. This produces 21,600 training and 5,400 test images for EuroSAT, and 1,680 training and 420 test images for UC Merced.

### B. Model Architectures

We selected eight architectures to cover three families. Table I lists them along with their parameter counts and publication years.

**Classical CNNs.** ResNet-50 and ResNet-101 are residual networks with 50 and 101 layers. DenseNet-121 connects each layer to every other in a feed-forward fashion, reusing features


Fig. 2. Sample images from EuroSAT. One randomly selected Sentinel-2 patch (64×64 pixels, 10 m resolution) is shown for each of the 10 classes.

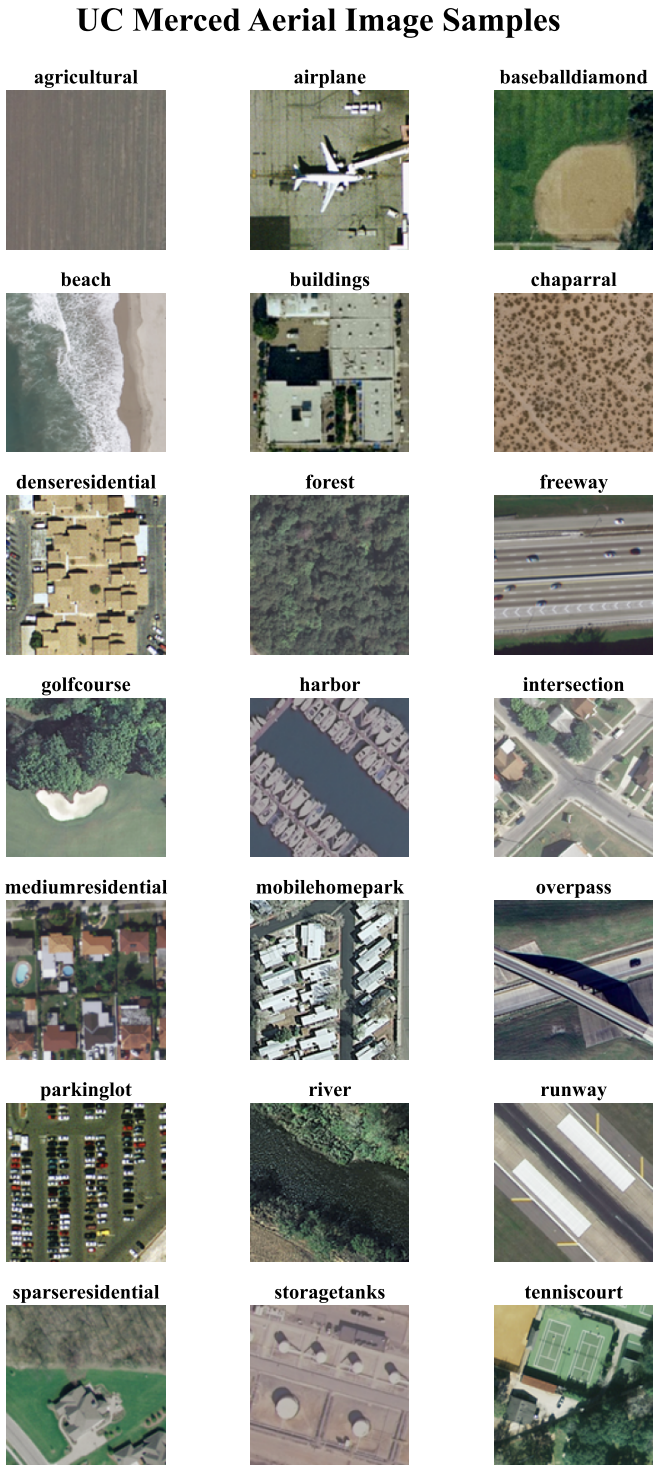


Fig. 3. Sample images from UC Merced. One randomly selected aerial image (256×256 pixels, 0.3 m resolution) is shown for each of the 21 classes.

TABLE I
MODEL ARCHITECTURES EVALUATED IN THIS STUDY. PARAMETER COUNTS REFER TO THE IMAGENET-PRETRAINED BACKBONE BEFORE REPLACING THE CLASSIFIER HEAD.

| Model | Family | Params (M) | Year |
|---|---|---|---|
| ResNet-50 [6] | CNN | 23.5 | 2016 |
| ResNet-101 [6] | CNN | 42.5 | 2016 |
| DenseNet-121 [7] | CNN | 7.0 | 2017 |
| EfficientNet-B0 [20] | CNN | 4.0 | 2019 |
| EfficientNet-B3 [20] | CNN | 10.7 | 2019 |
| ViT-B/16 [14] | Transformer | 85.8 | 2021 |
| Swin-T [15] | Transformer | 27.5 | 2021 |
| ConvNeXt-T [18] | Modern CNN | 27.8 | 2022 |

while keeping the parameter count at 7.0M. EfficientNet-B0 and B3 use depthwise separable convolutions with compound scaling and are the lightest and second-lightest models in our lineup.

**Vision Transformers.** ViT-B/16 divides each 224×224 image into 16×16 patches and processes them through 12 self-attention layers. At 85.8M parameters it is the largest model we test. Swin-T computes attention inside local windows that shift across layers, trading global receptive field for much lower memory use (27.5M parameters).

**Modernized CNN.** ConvNeXt-Tiny is a pure convolution network that borrows design choices from transformers: 7×7 kernels, LayerNorm, GELU activations, and an inverted bottleneck layout. It sits between the CNN and transformer families in both design philosophy and parameter count (27.8M).

All models start from ImageNet-1K pretrained weights loaded via the timm library [24]. We replace the final classification head with a new linear layer matching the target class count.

### C. Training Protocol

Every model is trained under an identical protocol to isolate architectural effects. All images are resized to 224×224 pixels and augmented with random horizontal/vertical flips, random rotation (±15°), and color jitter (brightness 0.2, contrast 0.2, saturation 0.1), followed by ImageNet normalization. We use AdamW [25] with a learning rate of $10^{-4}$ and weight decay of $10^{-4}$, a ReduceLROnPlateau scheduler (patience 5, factor 0.5), and early stopping with patience 10 on validation loss. Batch size is 32 and the maximum epoch count is 30. All training runs use PyTorch 2.0 [26] on an NVIDIA GPU with CUDA.

Fig. 4 shows the augmentation pipeline in action. The original image is on the left; two randomly transformed versions follow. These geometric and color perturbations expand the effective training set and reduce overfitting, especially for UC Merced where each class has only 80 training images.

### D. Evaluation Metrics

*1) Classification Metrics:* We report overall accuracy (OA), macro-averaged F1-score (unweighted mean across classes), and Cohen's kappa ($\kappa$) [27]. Kappa measures agreement beyond what chance would produce and is standard in remote sensing accuracy assessment [28], [29].

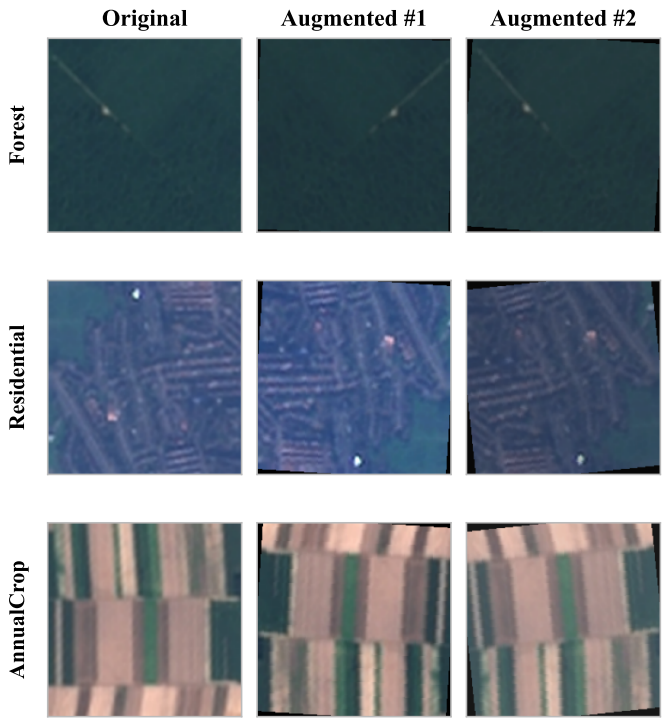


Fig. 4. Data augmentation pipeline. The leftmost column shows original EuroSAT images resized to 224×224 pixels. The two columns to the right show augmented versions produced by random flips, rotation, and color jitter.

*2) Statistical Significance:* We use McNemar's test [19], [30] with continuity correction to check whether accuracy differences between pairs of models are statistically real. The test statistic is:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{1}$$

where $n_{01}$ counts samples that model $A$ got right but $B$ got wrong, and $n_{10}$ the reverse. Under the null hypothesis of equal performance this follows a $\chi^2$ distribution with one degree of freedom. We use significance thresholds of $\alpha$ = 0.05 and 0.01.

*3) Computational Efficiency:* We record total trainable parameters and wall-clock training time for each model on each dataset.

## IV. RESULTS

### A. Overall Performance

Tables II and III list the classification results on both datasets. On EuroSAT, ConvNeXt-Tiny is first at 99.06%, followed by Swin-T (99.00%) and EfficientNet-B3 (98.98%). On UC Merced, EfficientNet-B3 leads at 99.76%, with three models tied at 99.52%.

What stands out is how small the gaps are. The entire range on EuroSAT is 0.93 percentage points (98.13% to 99.06%) and on UC Merced 0.95 points (98.81% to 99.76%). Every model exceeds 98% accuracy, which means that ImageNet pretraining brings all architectures to roughly the same performance floor regardless of their internal design. ResNet-101 finishes last on both datasets despite having 42.5M parameters, a point we return to in the Discussion.

TABLE II
CLASSIFICATION PERFORMANCE ON EUROSAT (5,400 TEST IMAGES). BEST RESULTS IN BOLD.

| Model | OA (%) | F1-Mac | $\kappa$ | Params |
|---|---|---|---|---|
| ConvNeXt-T | **99.06** | **0.9902** | **0.9895** | 27.8M |
| Swin-T | 99.00 | 0.9896 | 0.9889 | 27.5M |
| EffNet-B3 | 98.98 | 0.9894 | 0.9887 | 10.7M |
| ViT-B/16 | 98.91 | 0.9889 | 0.9878 | 85.8M |
| ResNet-50 | 98.81 | 0.9878 | 0.9868 | 23.5M |
| EffNet-B0 | 98.54 | 0.9853 | 0.9837 | 4.0M |
| DenseNet-121 | 98.46 | 0.9841 | 0.9829 | 7.0M |
| ResNet-101 | 98.13 | 0.9806 | 0.9792 | 42.5M |

TABLE III
CLASSIFICATION PERFORMANCE ON UC MERCED (420 TEST IMAGES). BEST RESULTS IN BOLD.

| Model | OA (%) | F1-Mac | $\kappa$ | Params |
|---|---|---|---|---|
| EffNet-B3 | **99.76** | **0.9976** | **0.9975** | 10.7M |
| EffNet-B0 | 99.52 | 0.9952 | 0.9950 | 4.0M |
| ViT-B/16 | 99.52 | 0.9952 | 0.9950 | 85.8M |
| ConvNeXt-T | 99.52 | 0.9953 | 0.9950 | 27.8M |
| ResNet-50 | 99.29 | 0.9929 | 0.9925 | 23.6M |
| DenseNet-121 | 99.29 | 0.9929 | 0.9925 | 7.0M |
| Swin-T | 99.05 | 0.9905 | 0.9900 | 27.5M |
| ResNet-101 | 98.81 | 0.9882 | 0.9875 | 42.5M |

Fig. 5 puts both datasets side by side in a bar chart. The bars cluster tightly near the top of the axis, which visually makes the point: these models are closer in performance than their very different designs would suggest.

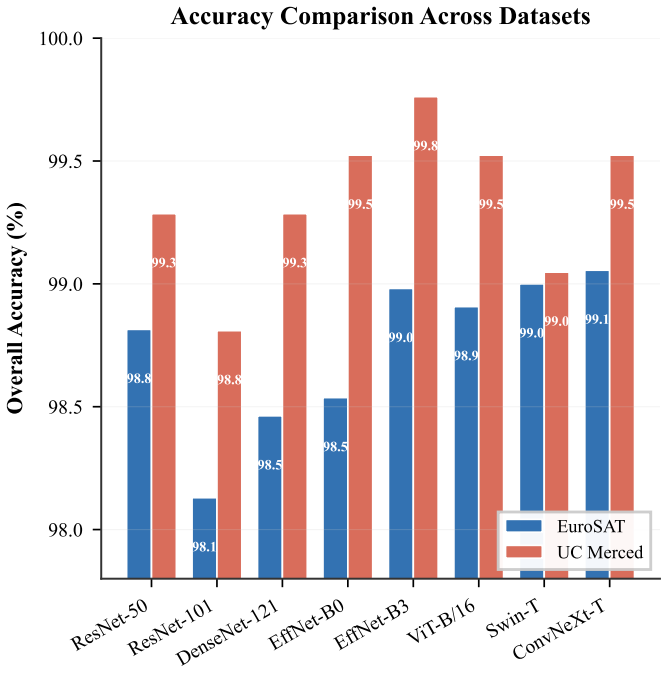


Fig. 5. Accuracy comparison across both datasets. All models exceed 98%, and the entire performance range spans less than one percentage point per dataset.

## B. Training Dynamics

Fig. 6 plots the training and test loss/accuracy curves for EuroSAT. Most models converge within 15 to 20 epochs, and early stopping kicks in before the 30-epoch limit for several of them. EfficientNet-B0 and DenseNet-121 converge fastest, reaching near-optimal accuracy in the first few epochs. ViT-B/16 and ResNet-50 take longer (best performance at epochs 27 and 29), which is consistent with their larger capacity needing more iterations to adapt.

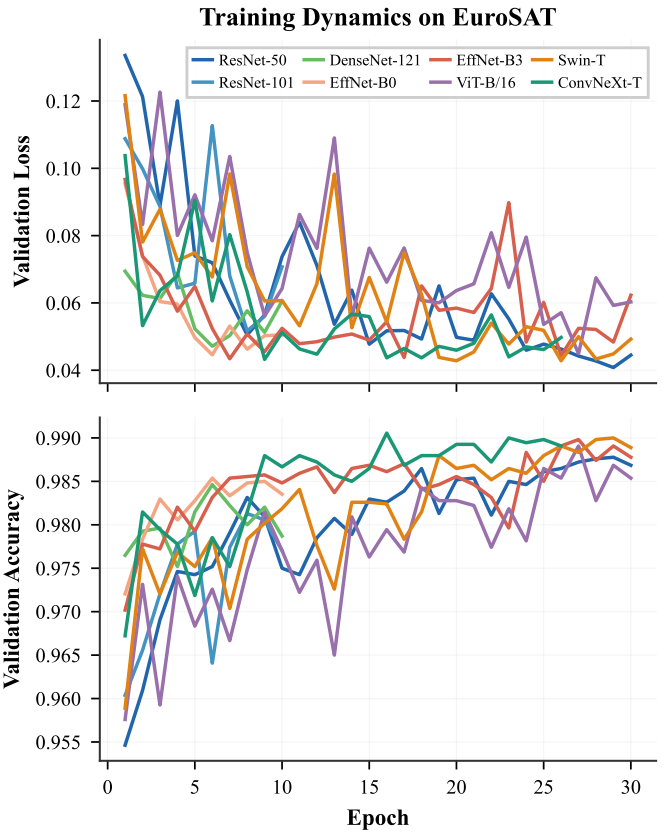


Fig. 6. Training dynamics on EuroSAT. Top: validation loss. Bottom: validation accuracy. Each line represents one architecture.

On UC Merced (Fig. 7), convergence is faster across the board, which makes sense given the smaller training set (1,680 images versus 21,600). With only 80 training images per class, fewer gradient updates are needed to adapt the pretrained features.

## C. Per-Class Analysis

*1) EuroSAT:* The per-class F1-score heatmap for EuroSAT (Fig. 8) shows that most classes are easy for every model. SeaLake and Forest both exceed 0.99 F1 across all eight architectures; their spectral signatures are distinct enough that no model struggles with them. PermanentCrop is the hardest class, with F1 values between 0.96 (ResNet-101) and 0.98 (ConvNeXt-Tiny). River also shows slightly more variation, probably because narrow river channels in 64×64 patches can look like roads.

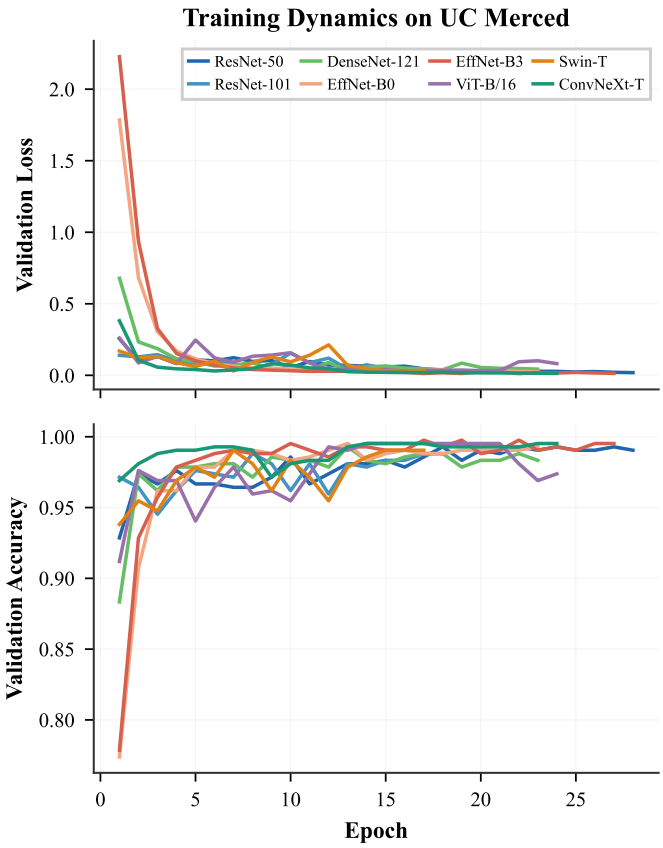


Fig. 7. Training dynamics on UC Merced. Top: validation loss. Bottom: validation accuracy. Convergence is faster due to the smaller dataset.

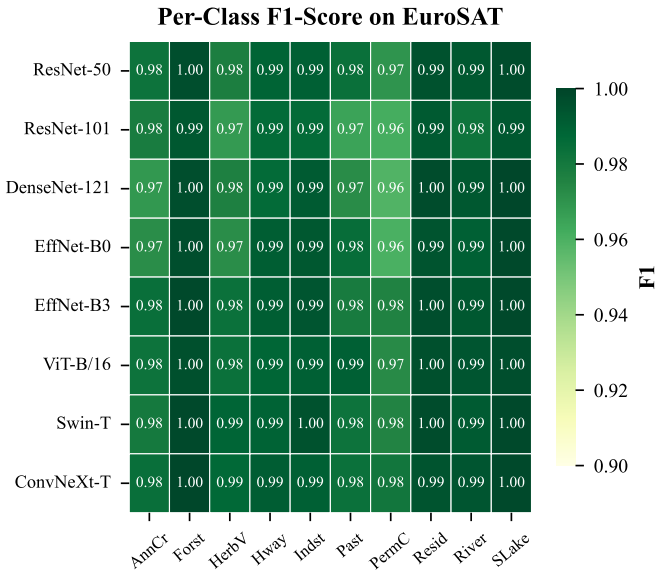


Fig. 8. Per-class F1-score heatmap on EuroSAT. Rows are models, columns are classes. Darker green is higher F1. PermanentCrop and River show the most variation.

*2) UC Merced:* On UC Merced (Fig. 9), the picture is even more uniform. Most cells in the heatmap are saturated at F1 = 1.0, meaning perfect classification. The exceptions are the residential classes: denseresidential, mediumresidential, sparseresidential, and to some extent buildings. These classes share similar visual content (rooftops, streets, trees), and the distinction between "dense" and "medium" residential is somewhat subjective even for a human interpreter.

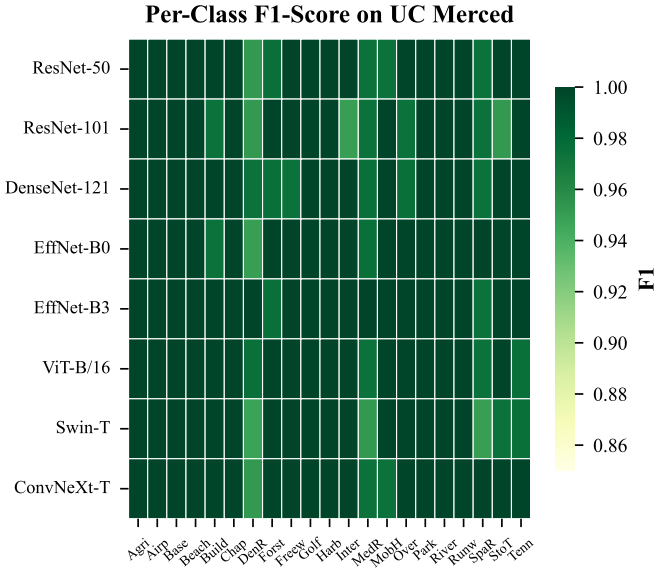


Fig. 9. Per-class F1-score heatmap on UC Merced (21 classes). Most cells are at F1 = 1.0 (dark green). Residential classes show the most variation.

### D. Statistical Significance

Fig. 10 shows the McNemar p-value matrix for EuroSAT. Out of 28 pairwise comparisons, only a handful produce $p < 0.05$, mostly involving ResNet-101 versus the better-performing models. The four top models (ConvNeXt-Tiny, Swin-T, EfficientNet-B3, ViT-B/16) are not significantly different from each other: the green cells in the upper-left block of the matrix show $p > 0.05$ for every pair. In other words, a different random test split could easily rearrange their ranking.

On UC Merced, even fewer pairs reach significance (not shown for brevity), which is expected given the smaller test set (420 images) and the tighter accuracy range. The McNemar results tell us that the accuracy ordering we observe is partly an artifact of which particular images ended up in the test set, not a reliable indicator of one architecture being strictly better than another.

### E. Error Analysis

Where do the remaining errors come from? Fig. 11 breaks down the predictions of ConvNeXt-Tiny on EuroSAT by class. Most classes have zero or near-zero misclassifications. The errors concentrate in Pasture and PermanentCrop: PermanentCrop gets confused with HerbaceousVegetation, and Pasture gets confused with AnnualCrop. Both patterns involve vegetation classes that share similar green tones at Sentinel-2 resolution.

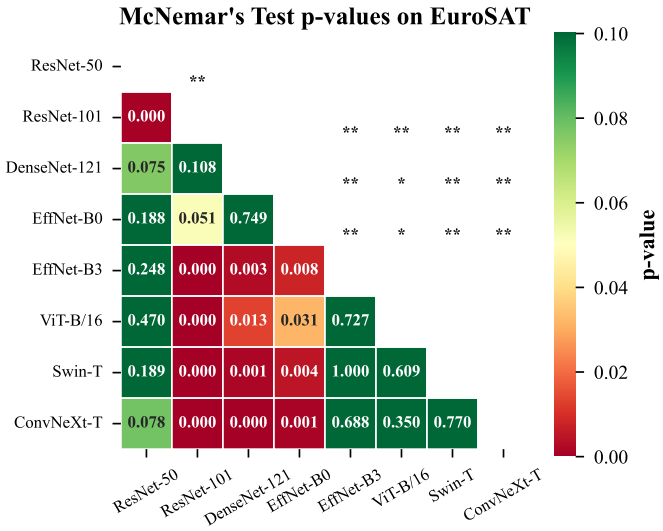


Fig. 10. McNemar's test p-value matrix for EuroSAT. Green = $p > 0.05$ (no significant difference); red = $p < 0.05$ (significant difference). Asterisks: * $p < 0.05$, ** $p < 0.01$.

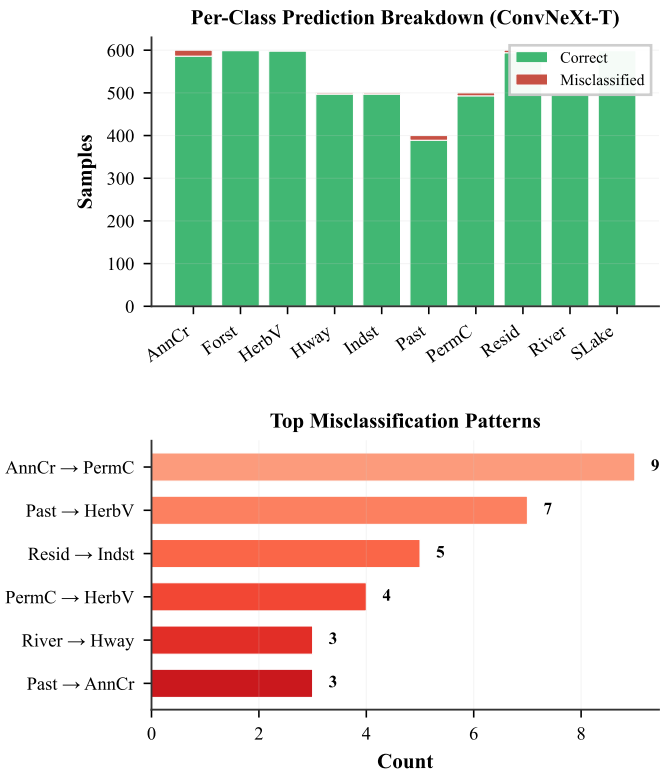


Fig. 11. Error analysis for ConvNeXt-Tiny on EuroSAT. Top: per-class correct (green) and misclassified (red) counts. Bottom: the most common misclassification patterns.

To see why these confusions happen, we traced the misclassified test images back to their source files and paired each one with a correctly classified example from the predicted class. Fig. 12 shows the result for EuroSAT, aggregating errors across all eight models. The left column (red border) is the misclassified image; the right column (blue border) is a typical example from the class the model predicted. The pairs are strikingly similar. A PermanentCrop patch with mixed crop rows and green margins looks almost identical to a HerbaceousVegetation patch at 64×64 pixels. Pasture fields with uniform grass coverage blend into AnnualCrop fields at this resolution. These are not model failures; they are cases where the images genuinely look the same.

Fig. 13 does the same for UC Merced. Even at the much finer 0.3 m resolution, some class boundaries remain blurry. Denseresidential and mediumresidential share the same building types and rooftop textures; the only real difference is how tightly the houses are packed, and in the worst cases even a human would hesitate. Mobilehomepark images contain scattered structures with surrounding green space that could easily pass for sparseresidential. The fact that these same pairs are confused by all eight architectures, not just one, confirms that the problem lies in the dataset definitions rather than in any particular model design.

### *F. Computational Efficiency*

Fig. 14 plots accuracy against parameter count for EuroSAT. EfficientNet-B0 sits in the bottom-left corner: 4.0M parameters, 98.54% accuracy. ViT-B/16 sits in the upper-right: 85.8M parameters, 98.91%. That is a 21× increase in model size for a 0.37 percentage point gain. The EfficientNet models trace out an efficiency frontier that no other family matches.

Table IV lists training times. EfficientNet-B0 is the fastest model on both datasets (344 s on EuroSAT, 289 s on UC Merced). ViT-B/16 is the slowest at 3,253 s on EuroSAT, consistent with the cost of computing global self-attention over 85.8M parameters. One surprise is ResNet-101: despite its large size (42.5M parameters), it trains in only 611 s on EuroSAT because early stopping terminates it after few epochs. The model converges quickly but then stalls, and the early stopping trigger fires before it has used its full training budget.

TABLE IV
TRAINING TIME COMPARISON (SECONDS). MODELS SORTED BY EUROSAT TIME.

| Model | EuroSAT | UC Merced |
|---|---|---|
| EfficientNet-B0 | 344 | 289 |
| DenseNet-121 | 516 | 315 |
| ResNet-101 | 611 | 259 |
| EfficientNet-B3 | 1,459 | 386 |
| Swin-T | 1,819 | 246 |
| ConvNeXt-Tiny | 2,032 | 379 |
| ResNet-50 | 2,807 | 369 |
| ViT-B/16 | 3,253 | 427 |

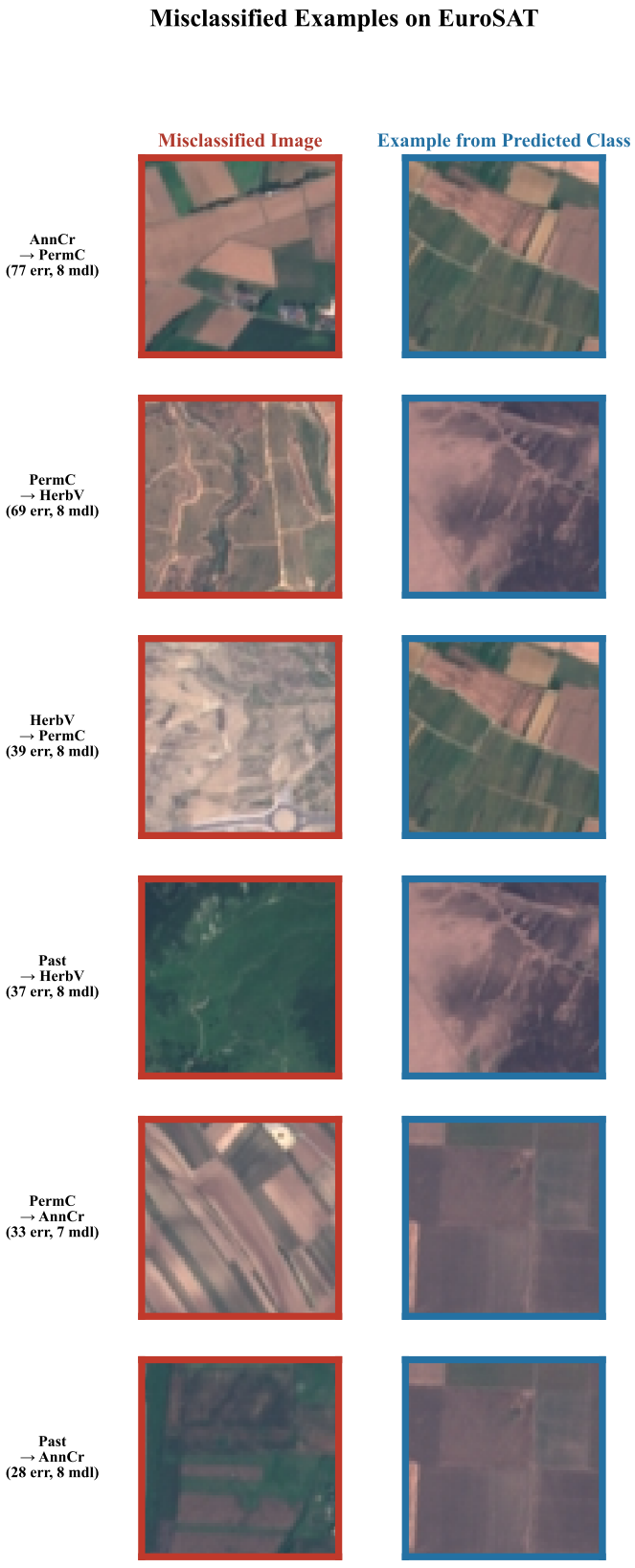


Fig. 12. Misclassified EuroSAT examples. Left (red border): misclassified test image. Right (blue border): correctly classified example from the predicted class. The visual similarity between each pair explains why models confuse them.

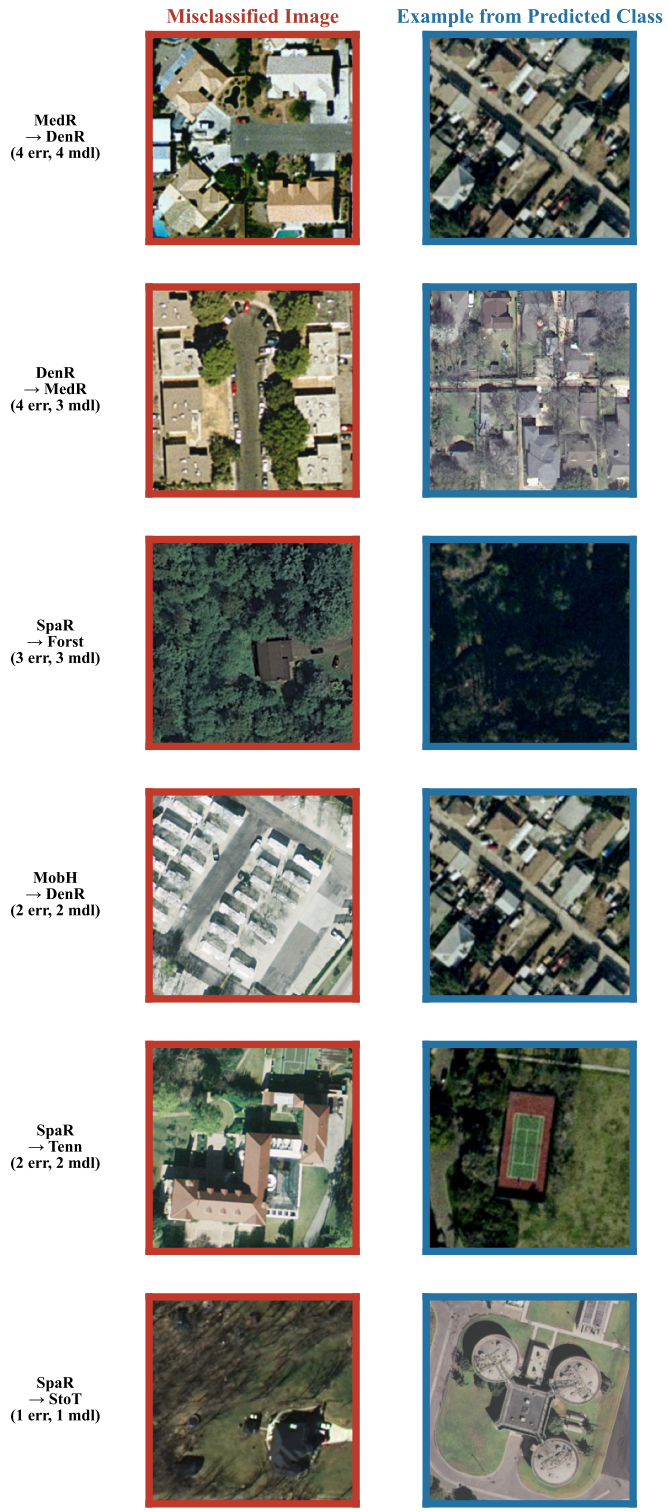


Fig. 13. Misclassified UC Merced examples. Same layout as Fig. 12. Residential density classes are the main source of confusion.

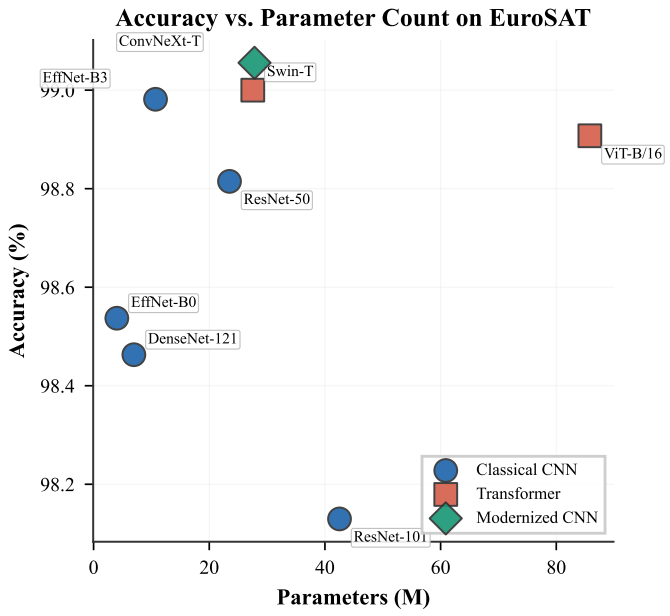


Fig. 14. Accuracy vs. parameter count on EuroSAT. Blue circles = classical CNNs, red squares = transformers, green diamonds = modernized CNN.

## V. Discussion

### A. Architecture Family Comparison

A common expectation is that transformers should outperform CNNs for scene classification because self-attention can model long-range spatial context. Our results tell a different story. On EuroSAT, a modernized CNN (ConvNeXt-Tiny) finished first. On UC Merced, a classical CNN (EfficientNet-B3) finished first. The transformers performed well, but they did not dominate. This is consistent with the original ConvNeXt paper [18], which argued that the accuracy gap between CNNs and transformers closes once CNNs adopt modern training practices.

Why did transformers not pull ahead? One possibility is the nature of the task. When images are resized to 224×224 pixels and each patch contains a single land-use type, local texture and color patterns may be enough for classification. Self-attention's ability to relate distant patches may become more useful for tasks that require understanding spatial layout, such as detecting a harbor by noticing boats near a dock, rather than just recognizing a uniform texture.

### B. The Depth Paradox

ResNet-101 consistently trailed ResNet-50 on both datasets (98.13% vs. 98.81% on EuroSAT, 98.81% vs. 99.29% on UC Merced), even though it has nearly twice as many parameters. This pattern, where a deeper pretrained network underperforms a shallower one after fine-tuning, has been reported before in transfer learning. The likely explanation is that 30 epochs of fine-tuning with early stopping is not enough to properly adapt all 101 layers. The deeper network starts from a good initialization but cannot move far from it before training stops, while the 50-layer version has a smaller parameter space that is easier to tune within the same budget.

### C. Parameter Efficiency

EfficientNet-B0 deserves special attention. With 4.0M parameters (18× fewer than ViT-B/16), it reaches 98.54% on EuroSAT and 99.52% on UC Merced. For anyone deploying a scene classification model on a mobile device, a drone, or an edge computing node, this is the clear choice. EfficientNet-B3, at 10.7M parameters, matches or beats every other model in our lineup while remaining small enough for practical deployment.

### D. Dataset Saturation

When the worst model in a benchmark still scores above 98%, the benchmark has arguably reached its useful limit. The narrow accuracy range we observed (less than one percentage point on both datasets) means that the difference between the "best" and "worst" architecture is within noise for most practical applications. This echoes calls in the community for harder evaluation scenarios: larger-scale datasets, cross-domain generalization tests, few-shot settings, and tasks that go beyond single-label classification [1], [4].

### E. Transfer Learning Dominance

The most striking pattern in our results is how little architecture matters once transfer learning is applied. The McNemar test (Fig. 10) shows that most accuracy differences are not statistically significant. Put differently: if we ran the same experiment with a different random test split, the model ranking would likely change. This implies that, at least for these benchmarks, the pretrained weights and the fine-tuning recipe are the main drivers of accuracy, not the architectural design itself. Whether this conclusion holds for more specialized tasks (multi-label classification, change detection, or few-shot learning) is an open question.

### F. Limitations

We tested on two datasets only; results could differ on larger or more diverse collections. We used only the RGB bands of EuroSAT, even though the full multispectral data might benefit some architectures more than others. We fixed all hyperparameters across models, which is fair for comparison but may not produce the best possible result for each individual architecture. And we used a single train-test split without cross-validation, so the results depend on one particular partition of the data.

## VI. Conclusion

We compared eight deep learning architectures across three design families on two remote sensing benchmarks and found that the differences between them are small. All models exceeded 98% accuracy with ImageNet pretraining and a uniform fine-tuning protocol. ConvNeXt-Tiny reached the top accuracy on EuroSAT (99.06%) and EfficientNet-B3 on UC Merced (99.76%), but McNemar's test showed that most pairwise gaps were not statistically significant. The smallest model, EfficientNet-B0 (4.0M parameters), came within one percentage point of every other model on both datasets.

For practitioners choosing a model, we recommend EfficientNet-B3 as a general default: it is small (10.7M parameters), fast to train, and competitive everywhere. If model size is the main constraint, EfficientNet-B0 is a strong lightweight alternative. Our results suggest that, for standard scene classification on current benchmarks, investing effort in training strategy and data quality is likely to matter more than switching to a fancier architecture. Future work should move to harder evaluation settings (larger datasets, cross-domain transfer, limited labels) where architectural differences may become more pronounced.

## REFERENCES


[1] G. Cheng, X. Xie, J. Han, L. Guo, and G.-S. Xia, "Remote sensing image scene classification meets deep learning: challenges, methods, benchmarks, and opportunities," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4697–4713, 2020.

[2] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, "Deep learning in remote sensing applications: a meta-analysis and review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019.

[3] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 270–279, 2010.

[4] 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.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[7] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," pp. 248–255, 2009.

[9] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognition*, vol. 61, pp. 539–556, 2017.

[10] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1264, 2018.

[11] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[12] M. Neumann, A. S. Pinto, X. Zhai, and N. Houlsby, "In-domain representation learning for remote sensing," *arXiv preprint arXiv:1911.06721*, 2019.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[16] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot, "SpectralFormer: rethinking hyperspectral image classification with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[17] D. Wang, Q. Zhang, Y. Xu, J. Zhang, and Y. Zhong, "Advancing plain vision transformer toward remote sensing foundation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–15, 2023.

[18] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 966–11 976.

[19] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[20] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[21] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[22] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: a novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[23] 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.

[24] R. Wightman, "PyTorch Image Models," 2019, gitHub repository, https://github.com/huggingface/pytorch-image-models.

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: an imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[27] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[28] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35–46, 1991.

[29] G. M. Foody, "Status of land cover classification accuracy assessment," *Remote Sensing of Environment*, vol. 80, no. 1, pp. 185–201, 2002.

[30] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.