Transfer Learning Architecture Selection for Remote Sensing Scene Classification

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

***Abstract*—This study quantifies the practical impact of architectural choice on remote sensing scene classification by evaluating eight deep learning models under strictly controlled conditions on EuroSAT (10 classes, 27,000 Sentinel-2 patches) and UC Merced (21 classes, 2,100 aerial photographs). The eight architectures represent three design paradigms: five classical convolutional networks (ResNet-50, ResNet-101, DenseNet-121, EfficientNet-B0, EfficientNet-B3), two attention-based vision transformers (ViT-B/16, Swin Transformer), and one hybrid-philosophy convolutional model (ConvNeXt-Tiny). All models share the same optimizer settings, the same augmentation pipeline, and the same ImageNet-1K pretrained starting point, meaning any accuracy difference comes from the architecture itself. 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

CLASSIFYING an entire satellite or aerial image patch into a single land-use category is one of the core problems in remote sensing, and accuracy on this task has jumped sharply since deep neural networks replaced traditional pipelines [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 hand-crafted features: color histograms, texture descriptors, and bag-of-visual-words representations [3]. Features were manually designed and fed to classifiers such as SVMs. While effective for simple cases, this approach was labor-intensive and did not scale well to large numbers of classes [4].

Convolutional neural networks changed the game. Architectures like VGGNet [5], ResNet [6], and DenseNet [7] learn multi-level features straight from pixels, so hand-designed descriptors are no longer needed. Training these networks first on the large ImageNet collection [8] and then fine-tuning them on remote sensing scenes beats the older pipelines by a wide margin [9], [10]. This works well because labeled satellite data is hard to come by, and transfer learning lets a model reuse what it already knows about natural images [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

He et al. [6] reframed each convolutional block as a residual mapping, where identity shortcuts bypass stacked layers so that the optimizer only learns the deviation from the input. This stabilized training beyond 100 layers and made the 50- and 101-layer variants the dominant backbones in remote sensing. Huang et al. [7] took a different approach by concatenating feature maps of all preceding layers as input to each subsequent layer, creating dense connectivity that maximizes feature reuse while keeping per-layer channel counts small; the result is a compact model that rivals much larger residual networks.

Tan and Le [20] showed that scaling depth, width, and input resolution together through one compound coefficient works better than enlarging any dimension on its own; their baseline network, found by neural architecture search, set a new bar for accuracy per FLOP on ImageNet. On the application side, Nogueira et al. [9] systematically compared full training, fixed feature extraction, and end-to-end fine-tuning across six architectures on three aerial datasets and concluded that adapting pretrained weights consistently surpasses learning from random initialization [2], [21].

B. Transformer-Based Approaches

Dosovitskiy et al. [14] dispensed with convolutions entirely by dividing an image into non-overlapping 16×16-pixel tiles, projecting each tile into a fixed-dimensional embedding, and processing the resulting token sequence through a multi-head self-attention encoder. Because every token can attend to every other, the model picks up spatial relationships that fixed-size kernels cannot see, which matters in remote sensing when relevant objects sit far apart in the image [17]. The downside is that memory grows quadratically with the number of tokens. Liu et al. [15] got around this by computing attention only inside small local windows and then cyclically shifting the window grid in the next layer so that neighboring windows exchange information. The cost drops to linear, and the network still builds multi-scale feature maps like a convolutional backbone does. Hong et al. [16] further extended the transformer paradigm to hyperspectral data, confirming that the architecture generalizes beyond three-channel inputs.

C. Modernized CNNs

Liu et al. [18] started from a plain ResNet and changed one design choice at a time to see how much each transformer idea was worth on its own: a patchify stem using non-overlapping 4×4 convolutions, depthwise 7×7 kernels that match the local window size of Swin, an inverted bottleneck with 4× expansion, and swapping batch normalization and ReLU for layer normalization and GELU. Self-attention was never added, yet the final model—still a pure convolution network—matched or beat Swin Transformer on ImageNet, COCO, and ADE20K. This is directly relevant to our work: if a convolution network can close the gap with a transformer just by borrowing its training recipe and a few design tweaks, the same thing might happen on remote sensing scenes.

D. Benchmark Datasets

Yang and Newsam [3] assembled the first publicly available high-resolution scene dataset by manually cropping 256×256 aerial patches from the USGS National Map Urban Area Imagery collection, yielding 100 images per class across 21 land-use categories at 0.3 m spatial resolution. Helber et al. [22] later introduced a medium-resolution counterpart built entirely from freely available Sentinel-2 acquisitions spanning 34 European countries; its 27,000 geo-referenced patches (64×64 pixels, 10 m ground sampling distance, 10 classes) made it the first large-scale scene benchmark derived from operational satellite data. Larger alternatives have since appeared—NWPU-RESISC45 [4] offers 31,500 images over 45 classes and AID [23] contains 10,000 images over 30 classes—yet EuroSAT and UC Merced persist as standard benchmarks because their manageable size permits exhaustive multi-model experiments within a single GPU budget. We picked these two because they sit at opposite ends of the spectrum: 0.3 m aerial photographs versus 10 m satellite imagery, fine-grained urban classes versus broad land-cover categories. If an architecture wins on both, the result is more convincing than a single-dataset comparison.

III. Methodology

Fig. 1 shows the experimental pipeline. Both datasets go through the same preprocessing, all eight architectures are trained with identical settings, and the resulting models are compared on accuracy, statistical significance, error patterns, and training cost.

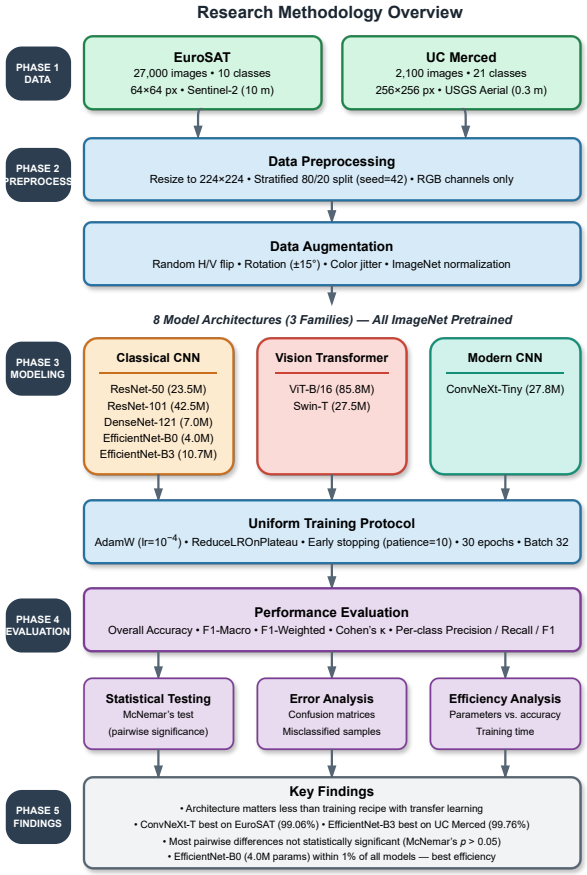

Fig. 1. Overview of the research methodology. Two benchmark datasets are preprocessed with a uniform pipeline and used to train eight architectures from three design families under identical conditions. The trained models are evaluated using classification metrics, statistical significance tests, error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

A. Datasets

1) EuroSAT: Our first evaluation dataset is EuroSAT [22], which comprises 27,000 geo-referenced image patches cap-

tured by the Sentinel-2 constellation and organized into 10 land-use categories. Individual patches measure 64×64 pixels with a 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 (left) shows one sample per class.

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, 256×256 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 both datasets appear in Fig. 2.

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.

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

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 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 with the same settings so that any accuracy difference must come from the architecture, not the optimizer or the augmentation. Images are resized to 224×224 pixels and augmented with random horizontal and vertical flips, rotation up to ±15°, and color jitter (brightness and contrast ±0.2, saturation ±0.1); after augmentation each channel is normalized to ImageNet mean and standard deviation. We optimize with AdamW [25] at a learning rate of 10^{-4} and weight decay of 10^{-4}. The learning rate is halved whenever validation loss stops improving for five epochs, and training stops entirely after ten epochs of no improvement. Batch size is 32, the maximum number of epochs is 30, and all runs use a single NVIDIA GPU with PyTorch 2.0 [26].

Fig. 3 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 three metrics for each model. Overall accuracy (OA) is simply the fraction of test images classified correctly. Macro-averaged F1-score averages the per-class F1 values without weighting by class size, so rare classes count as much as common ones. Cohen's kappa (κ) [27] corrects for the agreement that would occur by chance alone; it is a standard metric in remote sensing work because datasets often have uneven class distributions [28], [29].

2) Statistical Significance: To test whether two models really differ or just got lucky on different test images, we use McNemar's test [19], [30] with continuity correction. Dietterich showed this is the most reliable option when each classifier is run only once on a fixed test set. 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 χ^2 distribution with one degree of freedom. We use significance thresholds of α = 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%).

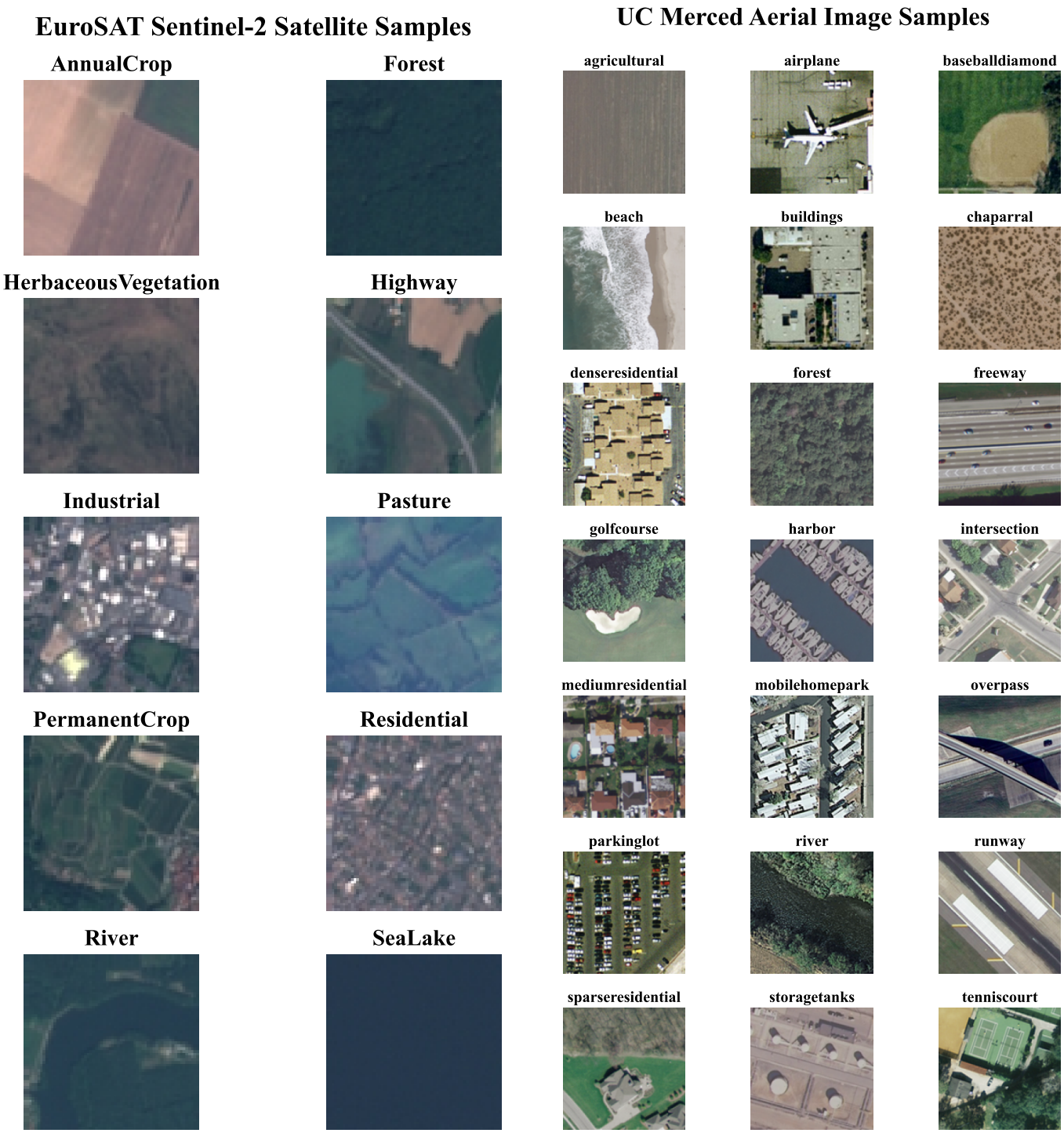

Fig. 2. Dataset samples. Left: one randomly selected Sentinel-2 patch per class from EuroSAT (64×64 pixels, 10 m resolution, 10 classes). Right: one randomly selected aerial image per class from UC Merced (256×256 pixels, 0.3 m resolution, 21 classes).

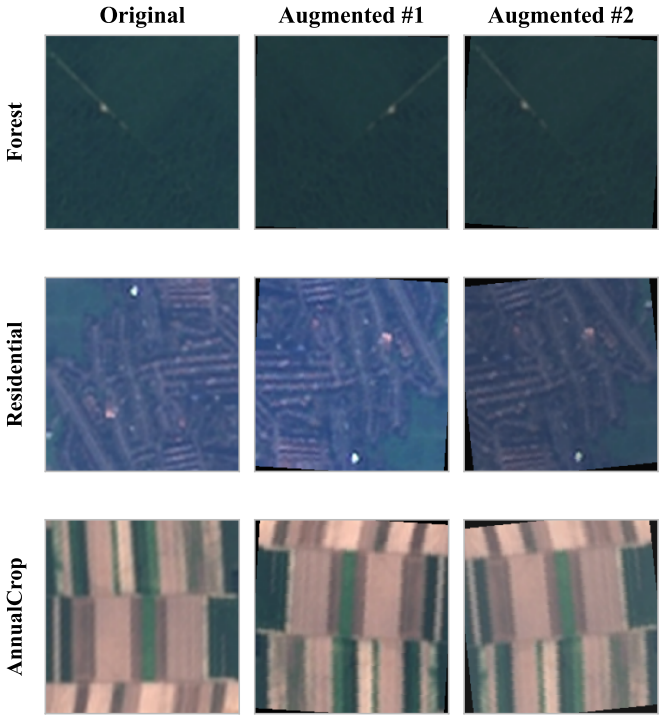

Fig. 3. Data augmentation pipeline. The leftmost column shows original EuroSAT images resized to 224×224 pixels. The two columns to the right show augmented versions produced by random flips, rotation, and color jitter.

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

Model	OA (%)	F1-Mac	κ	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

On UC Merced, EfficientNet-B3 leads at 99.76%, with three models tied at 99.52%.

The performance gaps are small. 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, as discussed further in Section V.

Fig. 4 presents both datasets side by side in a bar chart. The bars cluster tightly near the top of the axis, visually demonstrating that these models are closer in performance than their very different designs would suggest.

B. Training Dynamics

Fig. 5 plots the training and test loss/accuracy curves for EuroSAT. Most models converge within 15 to 20 epochs, and

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

Model	OA (%)	F1-Mac	κ	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

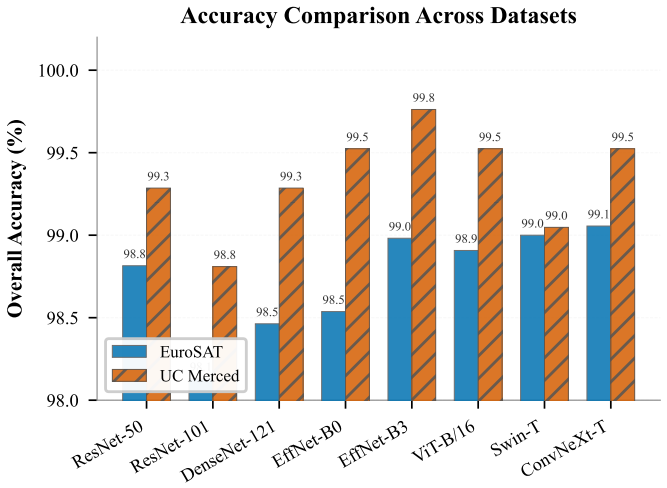

Fig. 4. Accuracy comparison across both datasets. All models exceed 98%, and the entire performance range spans less than one percentage point per dataset.

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.

On UC Merced (Fig. 6), 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. 7) 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.

2) UC Merced: On UC Merced (Fig. 8), the picture is even more uniform. Most cells in the heatmap are saturated at F1 = 1.0, meaning perfect classification. The exceptions are

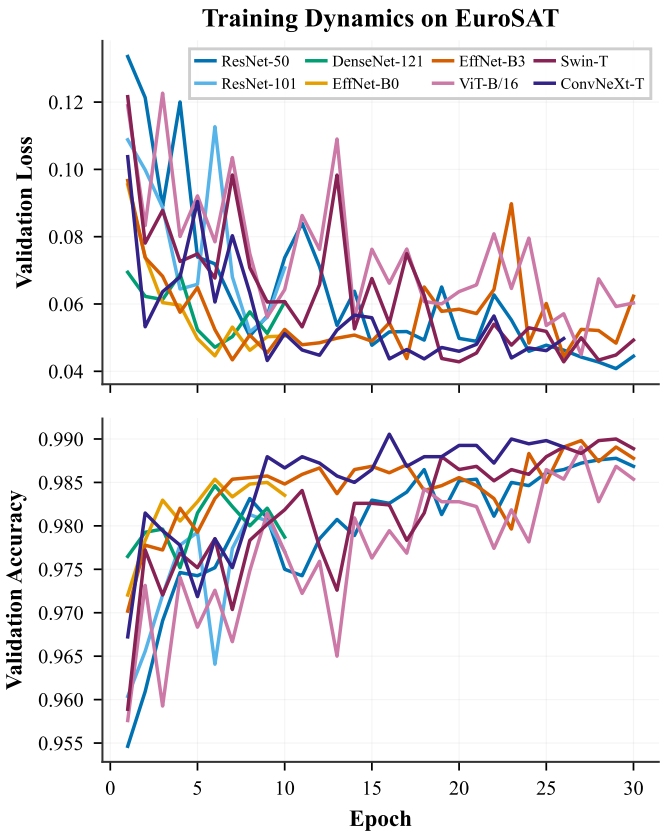

Fig. 5. Training dynamics on EuroSAT. Top: validation loss. Bottom: validation accuracy. Each line represents one architecture.

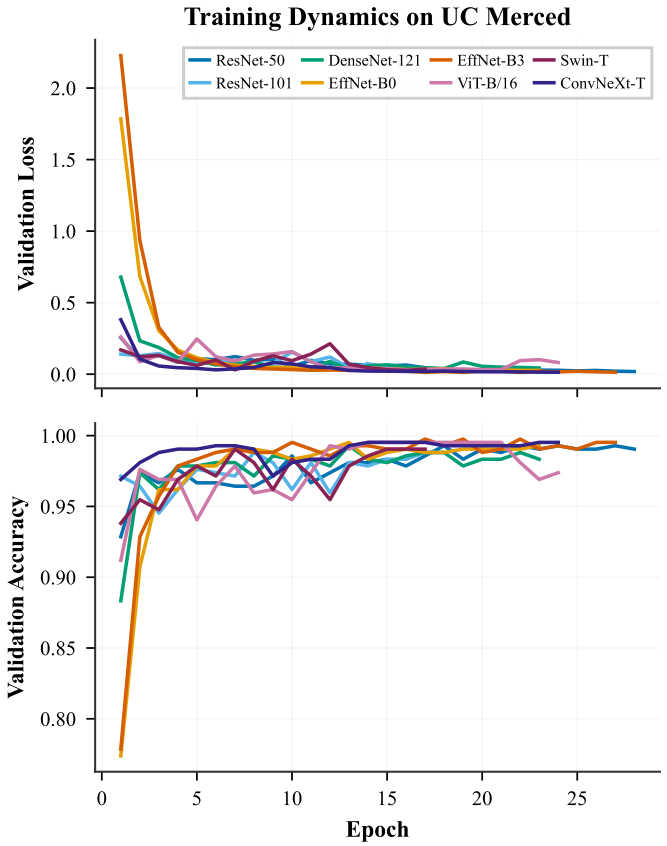

Fig. 6. Training dynamics on UC Merced. Top: validation loss. Bottom: validation accuracy. Convergence is faster due to the smaller dataset.

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.

D. Statistical Significance

Fig. 9 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. A different random split of the test set could easily shuffle 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

To identify the sources of remaining errors, Fig. 10 breaks down the predictions of ConvNeXt-Tiny on EuroSAT by

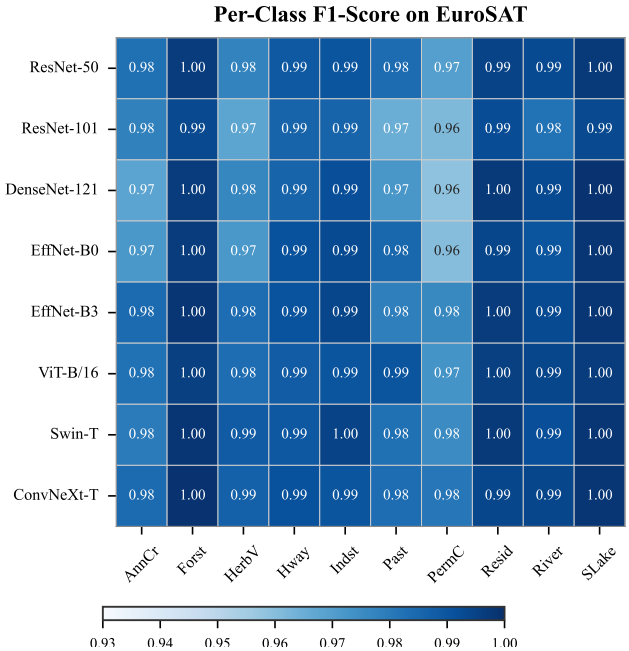

Fig. 7. Per-class F1-score heatmap on EuroSAT. Rows are models, columns are classes. Darker blue indicates higher F1. PermanentCrop and Pasture show the most variation across models.

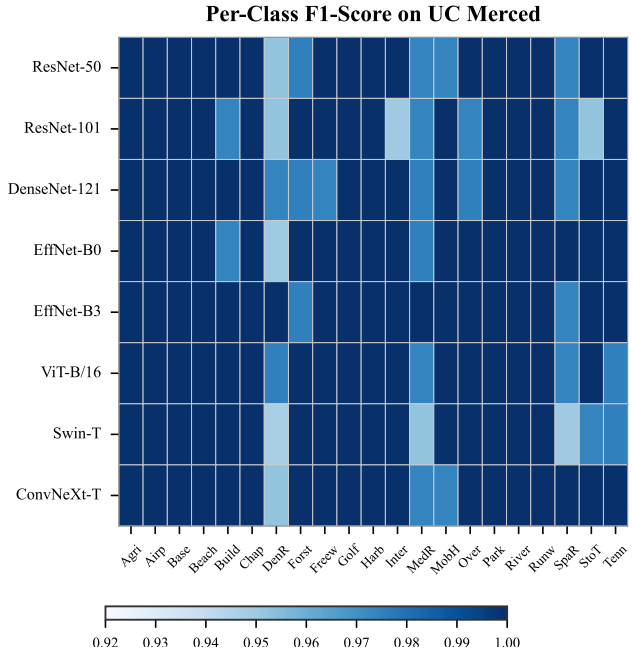

Fig. 8. Per-class F1-score heatmap on UC Merced (21 classes). Most cells are at F1 = 1.0 (dark blue). Residential sub-types show the most variation.

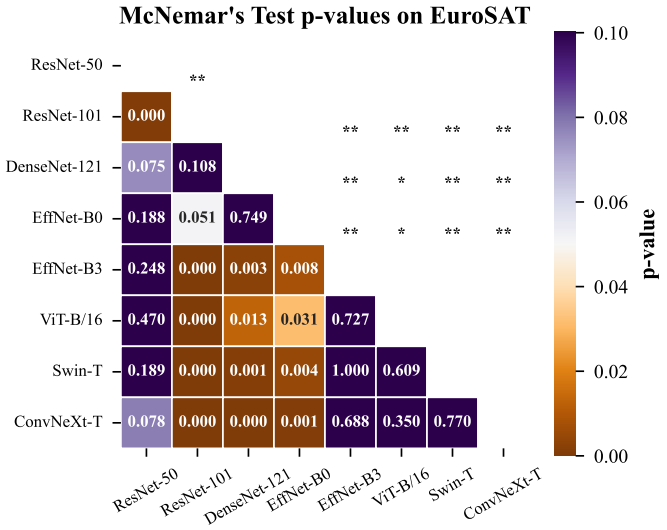

Fig. 9. McNemar's test p-value matrix for EuroSAT. Cells shaded green indicate model pairs whose accuracy difference is not statistically significant ($p > 0.05$); red cells denote pairs where the null hypothesis of equal error rate is rejected ($p < 0.05$). Significance levels: * $p < 0.05$, ** $p < 0.01$.

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.

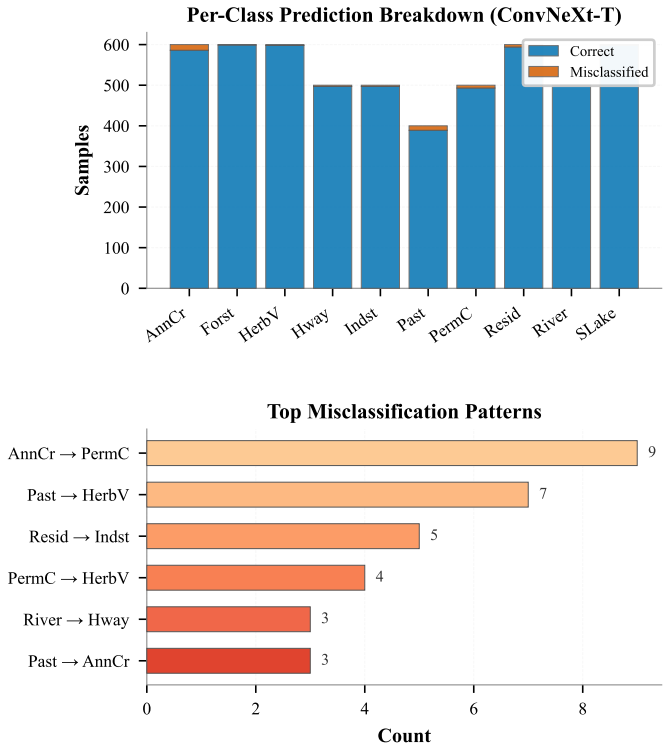

Fig. 10. Error analysis for ConvNeXt-Tiny on EuroSAT. Top: per-class correct (green) and misclassified (red) counts. Bottom: the most common misclassification patterns.

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. 11 (left) 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.

Even at the much finer 0.3 m resolution of UC Merced, 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. Fig. 11 shows representative pairs from both datasets.

F. Computational Efficiency

Fig. 12 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. ResNet-101 is the odd one out: 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
EffNet-B0	344	289
DenseNet-121	516	315
ResNet-101	611	259
EffNet-B3	1,459	386
Swin-T	1,819	246
ConvNeXt-T	2,032	379
ResNet-50	2,807	369
ViT-B/16	3,253	427

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. Liu et al. [18] made essentially the same point on ImageNet: once you give a convolution network the same training tricks that transformers use, the accuracy gap largely disappears.

The reason probably comes down to the task itself. At 224×224 pixels, each image shows one land-use type with fairly uniform texture, so local patterns in color and shape are enough to tell classes apart. Self-attention's ability to link distant parts of the image would matter more in scenes where spatial layout is the clue—say, recognizing a port because boats appear next to docks and water—but that kind of reasoning is not needed when the whole patch is "forest" or "highway."

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 stands out. 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 clearest takeaway is how little the architecture matters once transfer learning is in the picture. The McNemar test (Fig. 9) shows that most accuracy differences are not statistically significant. Running the same experiment with a different random test split would likely produce a different model ranking. At least on these two datasets, what you start with (the pretrained weights) and how you fine-tune matter more than which architecture you choose. Whether that still holds for 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.

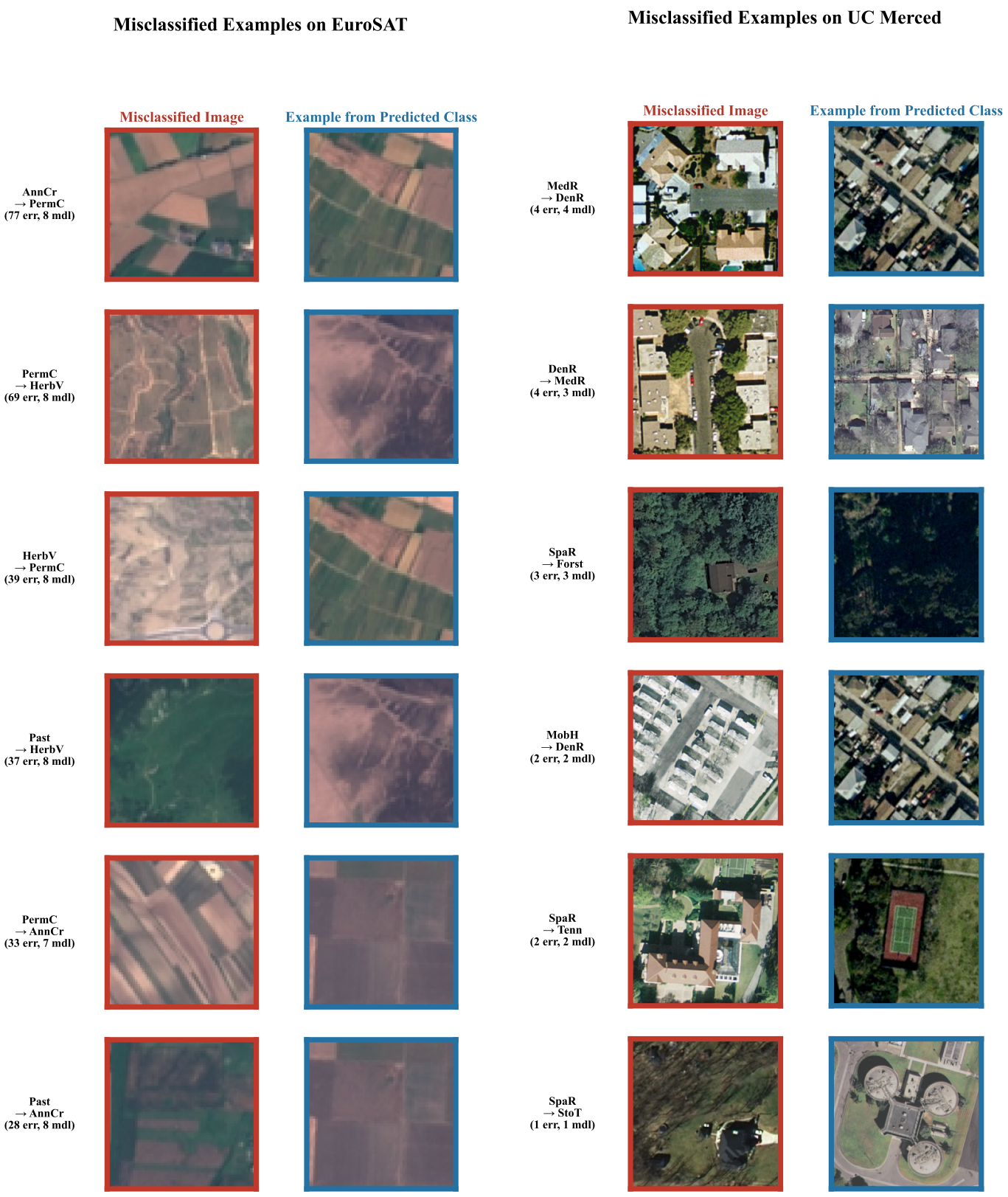

Fig. 11. Misclassified examples from EuroSAT (left) and UC Merced (right). In each pair, the red-bordered image was misclassified and the blue-bordered image is a correctly classified example from the predicted class. The visual similarity within each pair explains the confusion. On UC Merced, residential density classes are the main source of errors.

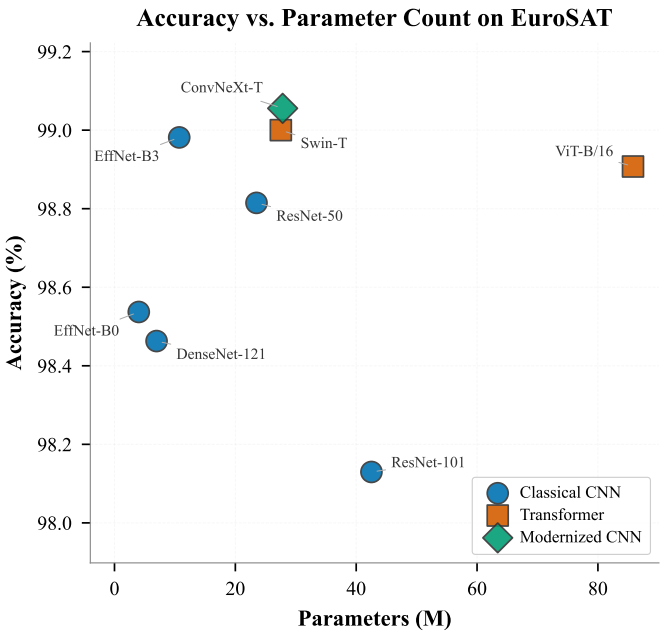

Fig. 12. Accuracy vs. parameter count on EuroSAT. Blue circles = classical CNNs, red squares = transformers, green diamonds = modernized CNN.

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. On these benchmarks, getting the training recipe right matters more than picking a fancier architecture. Future work should test on harder problems—bigger datasets, cross-domain transfer, and few-shot settings—where the gap between architecture families may actually show up.

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.