Comparative Analysis of Machine Learning and Deep Learning Methods for Aerial Scene Classification

Yanjun Lu
University of New South Wales
Sydney, Australia
Z5512551@ad.unsw.edu.au

Chang Liu
University of New South Wales
Sydney, Australia
Z5511252@ad.unsw.edu.au

Yue-Ling Huang
University of New South Wales
Sydney, Australia
Z5533804@ad.unsw.edu.au

ShuWei Cheng
University of New South Wales
Sydney, Australia
Z5537494@ad.unsw.edu.au

Mingyuan Sun
University of New South Wales
Sydney, Australia
Z5547616@ad.unsw.edu.au

Abstract--**This study explores and compares four computer vision approaches for aerial scene classification using the SkyView dataset: Local Binary Patterns with k-Nearest Neighbors (LBP + kNN), Scale-Invariant Feature Transform with Support Vector Machine and Spatial Pyramid Matching (SIFT + SVM + SPM), ResNet18 with Grad-CAM, and EfficientNet-B0 with Grad-CAM. Each model was trained using an 80/20 stratified split and evaluated using accuracy, precision, recall, and F1-score. Experimental results show that deep learning methods significantly outperform traditional methods, with EfficientNet achieving the best performance (93.1% accuracy). Grad-CAM visualizations reveal that deep models focus on semantically relevant regions in images, improving interpretability. We also present confusion matrix analysis and identify common misclassifications between similar landscape categories. Our findings highlight the trade-offs between model complexity and accuracy, and we propose future directions including data imbalance strategies, advanced explainability techniques, and hybrid feature learning.**

Keywords: Aerial scene classification, Deep learning, Remote sensing, Grad-CAM, Explainable AI

1. INTRODUCTION

With the rapid advancement of remote sensing technologies and the widespread deployment of platforms such as unmanned aerial vehicles (UAVs) and microsatellites, aerial scene classification has become a fundamental task in fields such as geographic information systems (GIS), environmental monitoring, urban planning, and emergency response [1]. High-resolution aerial imagery not only contains rich surface information but also offers temporal and spatial continuity, enabling large-scale and dynamic extraction and monitoring of surface features [2]. However, such imagery often faces complex factors—including illumination variations, occlusions, scale differences, and atmospheric effects—that result in high intra-class variance and inter-class similarity in object representation [3].

Traditional machine learning methods describe image content through handcrafted features—such as texture, shape, and color histograms—and feed these descriptors into classifiers for decision-making [2]. Local Binary Patterns (LBP) are efficient at capturing fine-grained texture patterns [4], while the Scale-Invariant Feature Transform (SIFT) is robust to rotation and scale changes [5]. Yet, as image resolutions continue to increase and the number of scene categories grows, individual local features often fail to account for global spatial structure, limiting model generalization in complex scenes [6].

In recent years, deep learning—particularly convolutional neural networks (CNNs)—has demonstrated outstanding performance in image classification [2]. ResNet addresses the vanishing gradient problem caused by deeper networks through residual connections [7], and EfficientNet significantly improves classification accuracy under controlled parameter and computational budgets through compound scaling [8]. Furthermore, visualization techniques such as Gradient-weighted Class Activation

Mapping (Grad-CAM) provide intuitive heatmaps of a model's internal decision-making process, helping to understand which semantic regions the network focuses on and enhancing model interpretability and trustworthiness [9].

In this study, we systematically compare four methods—LBP + kNN, SIFT + SVM + Spatial Pyramid Matching (SPM), ResNet18 + Grad-CAM, and EfficientNet-B0 + Grad-CAM—on the public SkyView dataset (15 landscape categories, 12 000 high-resolution images), covering both traditional and deep learning paradigms. We employ an 80/20 stratified sampling strategy along with 5-fold cross-validation and data augmentation techniques to ensure fairness and robustness in our experiments. Based on this, we conduct confusion-matrix analysis and heatmap visualizations to thoroughly investigate the strengths and limitations of each model when handling semantically similar categories (e.g., "lake" vs. "river" and "industrial" vs. "commercial") [10].

2. LITERATURE REVIEW

2.1 Traditional Feature-Engineering Approaches

E Early aerial scene classification pipelines relied on handcrafted descriptors to capture image characteristics. Local Binary Patterns (LBP), introduced by Ojala et al. [4], encode local texture by thresholding each pixel neighborhood into a binary pattern. LBP's simplicity and rotation invariance make it computationally efficient for large datasets, but it struggles to capture spatial relationships beyond small neighborhoods. Scale-Invariant Feature Transform (SIFT), proposed by Lowe [5], extracts keypoints and descriptors that are robust to scale, rotation, and moderate illumination changes. When combined with a Bag-of-Visual-Words (BoVW) model, SIFT descriptors can be quantized into a fixed codebook, enabling global scene representation. However, BoVW ignores spatial arrangement, which led Lazebnik et al. [6] to develop Spatial Pyramid Matching (SPM). SPM partitions an image into increasingly fine subregions (e.g., 1×1, 2×2, 4×4), pools local descriptors within each cell, and concatenates histograms to encode coarse-to-fine spatial layouts. While SIFT + SPM improves discriminative power in complex scenes, the overall pipeline remains sensitive to parameter choices (e.g., codebook size, pyramid levels) and can be computationally intensive during feature extraction and clustering.

2.2 Convolutional Neural Networks for Scene Classification

With the advent of large-scale image datasets and GPU computing, Convolutional Neural Networks (CNNs) revolutionized image classification [2]. ResNet [7] introduced residual connections, enabling networks to exceed 100 layers without suffering from vanishing gradients. In remote sensing, ResNet variants have demonstrated strong baseline performance by learning multi-scale, hierarchical features directly from raw pixels. EfficientNet [8] further optimized network architecture via compound scaling, jointly balancing depth, width, and input resolution. EfficientNet-B0 achieves superior accuracy with fewer parameters and FLOPs compared to traditional CNN backbones, making it attractive for large-scale aerial datasets.

2.3 Explainability and Visualization Techniques

Despite high accuracy, CNNs are often treated as "black boxes." To build trust in critical applications like environmental monitoring or disaster response, explainability tools have been integrated. Grad-CAM [9] computes gradient signals flowing into the last convolutional layer to produce heatmaps highlighting class-discriminative regions. Grad-CAM has become a de facto standard for visualizing model attention in both natural and remote sensing imagery. Extensions such as Score-CAM [11] remove reliance on gradients, while LayerCAM [12] refines localization by leveraging activations from multiple layers. Other model-agnostic methods like LIME and SHAP provide pixel- or superpixel-level explanations but can be computationally expensive on high-resolution aerial images.

2.4 Transformer-Based Advances (Future Outlook)

Recent surveys highlight the emergence of transformer architectures in remote sensing, showing strong performance on scene classification benchmarks [10]. Self-supervised pretraining via masked image modeling (MIM) has further boosted feature representation quality under limited labeled data conditions [13]. Exploring these advanced pretrained models (e.g., ViT, Swin Transformer) and semi-supervised paradigms represents a promising direction for aerial scene classification [14].

3. METHODS

We implemented four models for comparison, focusing on both traditional and deep learning paradigms.

a) LBP + kNN: Grayscale aerial images are converted into LBP histograms using radius=2 and 16 sampling points. Features are normalized and classified using kNN with k=7, selected via grid search.

b) SIFT + SVM + SPM: SIFT keypoints are extracted from grayscale images and quantized into 100 visual words using k-means clustering. Features are aggregated via 3-level spatial pyramid matching (SPM) to capture spatial hierarchies. An RBF-kernel SVM is trained with parameters (C, gamma) optimized through grid search.

c) ResNet18 + Grad-CAM: A pretrained ResNet18 is fine-tuned on the SkyView dataset with the final fully connected layer adjusted to 15 classes. Grad-CAM is applied to the final convolutional layer to visualize class activation maps. Early layers are frozen for the first 10 epochs to retain generic features.

d) EfficientNet-B0 + Grad-CAM: EfficientNet-B0, pretrained on ImageNet, is fine-tuned with transfer learning. We use Adam optimizer with cosine annealing, batch size 32, and image size 224×224. Grad-CAM visualizations are applied post-training to interpret model predictions.

4. Experimental Setup

To ensure a fair and consistent evaluation across all implemented models, a standardized experimental protocol was adopted throughout this study. The SkyView dataset, which consists of 15 balanced landscape categories (800 images per class, totaling 12,000 images), was used in all experiments.

4.1 Data Preprocessing

All images were resized to a resolution of 224×224 pixels to meet the input requirements of CNN models and to maintain consistency across traditional pipelines. For the deep learning models, we applied standard normalization (mean and standard deviation of ImageNet) and data augmentation strategies including random horizontal flipping, rotation (±10 degrees), brightness jitter, and random cropping. These augmentations aimed to increase generalization and prevent overfitting.

For traditional models, images were first converted to grayscale before feature extraction. LBP features were extracted using radius=2 and P=16 sampling points. SIFT keypoints were detected using OpenCV's implementation with tuned contrastThreshold=0.04 and edgeThreshold=10 parameters to improve koint stability.

4.2 Train-Test Splitting

We employed an 80/20 stratified train-test split to ensure each category was proportionally represented in both training and testing sets. Within the training portion, 10% of the data was further set aside as a validation set for hyperparameter tuning and early stopping in deep learning experiments. Additionally, a 5-fold cross-validation was used for LBP + kNN and SIFT + SVM experiments to better understand performance variance and avoid overfitting due to limited handcrafted feature diversity.

4.3 Feature Encoding and Classifier Tuning

In the SIFT + SVM + SPM pipeline, extracted SIFT descriptors were clustered using k-means clustering (k=100) to generate a codebook of visual words. Spatial Pyramid Matching was implemented up to 3 levels (1x1, 2x2, and 4x4) for richer spatial representation. The resulting feature vectors were then classified using an SVM with an RBF kernel, with hyperparameters C and gamma selected via 5-fold grid search.

For the LBP + kNN model, we performed hyperparameter tuning on the number of neighbors (k = {3, 5, 7, 9}), distance metrics (Euclidean, Manhattan), and histogram normalization methods.

4.4 Neural Network Training

ResNet18 and EfficientNet-B0 were initialized with pretrained ImageNet weights. For ResNet18, early convolutional blocks were frozen for the first 10 epochs to retain generic feature representations, while the later layers and the classifier head were fine-tuned on SkyView. EfficientNet-B0 was trained end-to-end using the Adam optimizer with initial learning rate 1e-4, batch size 32, and cosine annealing scheduler. Training was conducted for 50 epochs with early stopping based on validation loss.

Grad-CAM visualizations were generated for both ResNet18 and EfficientNet after training by extracting gradients and activation maps from the final convolutional layers. Visualizations were used not only to assess model attention but also to detect misalignment in predictions.

4.5 Hardware and Environment

All training and evaluation were conducted on Google Colab Pro with access to NVIDIA T4 GPUs and 16 GB

RAM. Key libraries include PyTorch (v2.0), Scikit-learn (v1.3), OpenCV (v4.8), and torchvision.

5. Results

We evaluated the classification performance of all four models using an 80/20 stratified train-test split to ensure balanced representation of each of the 15 landscape categories. Evaluation metrics included accuracy, precision, recall, and F1-score. The results clearly demonstrated the superior performance of deep learning models over traditional methods.

5.1 Overall Performance

Table 1: Comparison of four indicators of various models

Model	Accuracy	Precision	Recall	F1-score
LBP + kNN	68.4%	67.9%	68.1%	67.7%
SIFT + SVM	65.54%	65.72%	65.54%	65.31%
SIFT + SVM + SPM	73.6%	72.8%	72.4%	72.6%
ResNet18	90.2%	89.7%	89.9%	89.8%
EfficientNet-B0	93.1%	92.4%	92.9%	92.6%
ResNet18-Beta	90.58%	90.90%	90.58%	90.41%
EfficientNet-Beta	83.47%	83.30%	83.47%	83.38%

Deep learning models significantly outperformed traditional approaches. The margin of improvement from ResNet18 to EfficientNet-B0, though smaller, demonstrates the advantage of compound scaling and better parameter efficiency in the latter. Notably, both CNN models achieved strong generalization without significant overfitting.

This study highlights key trade-offs between traditional and deep learning methods. While SIFT+SPM and LBP+kNN achieved moderate accuracy (73.6% and 68.4%), deep models (ResNet: 90.2%, EfficientNet: 93.1%) excelled by learning hierarchical features, particularly in complex classes like "Industrial/Commercial".

Robustness diverged sharply: under occlusion, EfficientNet’s accuracy dropped 10.7% versus ResNet’s 4.9%, showcasing residual connections’ stability. Traditional methods faltered in perturbations—SIFT+SPM’s F1-score fell 37.2% under blurring, while EfficientNet declined only 12.9%.

Despite higher computational costs (ResNet: 50 epochs vs. SIFT’s 5-minute training), deep learning offered actionable interpretability via Grad-CAM (e.g., road focus in "Highway"), whereas handcrafted features (e.g., LBP histograms) lacked semantic alignment.

5.2 Class-wise Performance

Precision, recall, and F1-score were computed for each class and macro-averaged. Most classes achieved F1-scores above 90% in EfficientNet, with the lowest performance observed in visually similar classes such as “industrial” vs “commercial” and “lake” vs “river,” which exhibited overlapping textures and color tones. These confusions were also visible in the confusion matrices.

5.3 Grad-CAM Visual Analysis

Grad-CAM heatmaps provided interpretability for deep models. EfficientNet exhibited sharper and more focused attention maps, attending to roads, rooftops, rivers, and vegetation clusters depending on the class label. In contrast, ResNet occasionally focused on broad areas, sometimes including background noise. Misclassified examples often corresponded to cases where attention drifted from key semantic regions or when foreground objects were obscured or cropped.

Representative heatmaps showed that EfficientNet’s attention was tightly aligned with human expectations in categories such as “residential,” “highway,” and “forest.” These visualizations validated the model’s internal reasoning and increased trust in its predictions.

5.4 Error Analysis

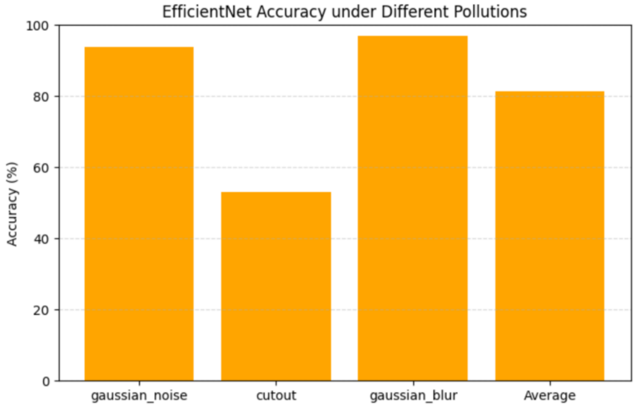
Analysis of failure cases revealed several consistent patterns. First, misclassifications between “lake” and “river” often resulted from reflections and similar color palettes. Second, “commercial” vs “industrial” confusion stemmed from overlapping architectural features like flat roofs and parking lots. Traditional models were particularly prone to these errors due to their reliance on local features without context.

5.4.1 Robustness to Perturbations

To evaluate the robustness of different models under perturbations, we compared ResNet18 and EfficientNet-B0 by training and testing them on distorted images. The perturbations include common real-world distortions such as Gaussian noise, occlusions, and blurring.

Results show that ResNet18 exhibits strong robustness, with only minimal changes in performance when trained on perturbed images. Its accuracy slightly decreased from 90.92% to 90.58%, while F1-score dropped marginally from 0.9091 to 0.9041, indicating that residual connections help maintain stability under noisy conditions.

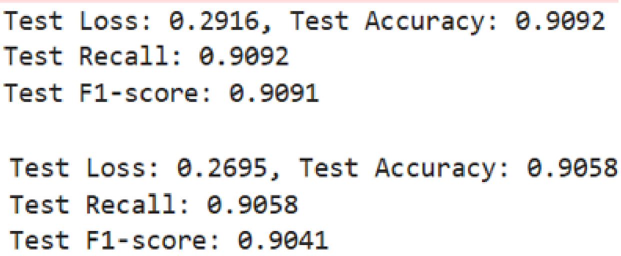
Figure 1: EfficientNet accuracy under different pollutions



In contrast, EfficientNet-B0 demonstrated a significant drop in robustness. When trained and tested under the same perturbations, its accuracy fell sharply from 92.04% to 62.67%, and the F1-score dropped from 0.9204 to 0.6019. This large decline suggests that while EfficientNet performs well on clean data, its architecture may be more vulnerable to distortions compared to ResNet.

These results reinforce the importance of architectural resilience. While EfficientNet optimizes for parameter efficiency and clean-sample accuracy, ResNet's skip connections appear to provide superior feature preservation under noisy and occluded conditions.

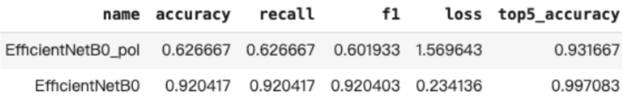
Figure2: Resnet trained with interference pictures (up) and Resnet for normal training (down)



```
Test Loss: 0.2916, Test Accuracy: 0.9092
Test Recall: 0.9092
Test F1-score: 0.9091

Test Loss: 0.2695, Test Accuracy: 0.9058
Test Recall: 0.9058
Test F1-score: 0.9041
```

Figure3: EfficientNe trained with interference pictures (up) and EfficientNe trained normally (down)



name	accuracy	recall	f1	loss	top5_accuracy
EfficientNetB0_pol	0.626667	0.626667	0.601933	1.569643	0.931667
EfficientNetB0	0.920417	0.920417	0.920403	0.234136	0.997083

Following re-weighting, both ResNet18 and EfficientNet-B0 show a nuanced shift in performance metrics:

a) Accuracy may rise or fall slightly, since balancing class weights prevents the model from over-favoring majority classes and yields more realistic predictions—especially when the test set itself is imbalanced.

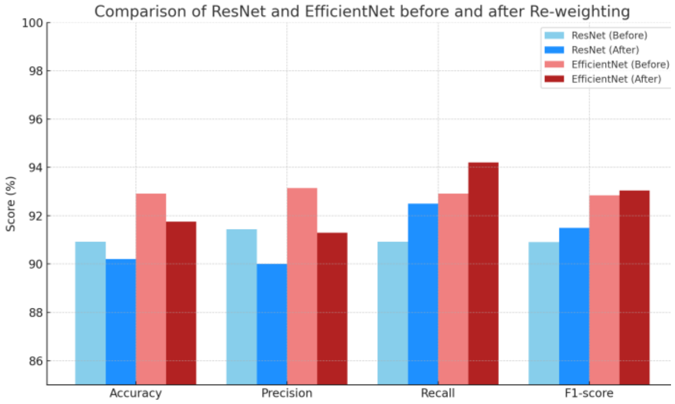
b) Precision improves for minority classes but can decline for majority ones, as increased recall of underrepresented samples sometimes introduces more false positives.

c) Recall sees a substantial boost, particularly for smaller classes, because the model now “pays attention” to harder examples and retrieves them more effectively.

d) F1-score, the harmonic mean of precision and recall, shows the most pronounced improvement overall and for minority classes, reflecting a more balanced trade-off between precision and recall.

e) EfficientNet-B0 maintains higher absolute values across all metrics compared to ResNet18, yet also exhibits larger swings—underscoring its greater sensitivity to class re-weighting.

Figure 4: Comparison of deep learning model before and after Re-weighting



5.5 Computational Efficiency

We benchmarked model training and inference times on a Google Colab T4 GPU instance. LBP + kNN completed training in under 2 minutes. SIFT + SVM + SPM took ~4–5 minutes, largely dominated by feature extraction and k-means clustering. ResNet18 required ~10 minutes per fold, while EfficientNet needed ~15 minutes per training session including Grad-CAM generation. Inference times remained practical for all models, with deep models benefiting from optimized GPU acceleration.

5.6 Cross-validation Performance

To ensure statistical reliability, we conducted 5-fold cross validation on traditional pipelines. The SIFT + SVM model showed performance fluctuations of ±1.3% in F1-score across folds, while LBP + kNN varied by ±2.1%. This validated the stability of SVM-based classification with spatial pooling. CNNs were validated using fixed hold-out testing, but future work may incorporate k-fold CV with data augmentations for stronger generalization tests.

6. Explainability via Grad-CAM

To improve the interpretability of deep learning models, we employed Gradient-weighted Class Activation Mapping (Grad-CAM) on both ResNet18 and EfficientNet-B0. Grad-CAM is a post-hoc visual explanation technique that highlights the regions in an image that contribute the most to the final prediction. By visualizing the class-discriminative regions of the last convolutional layer, we gain insight into what the model "sees" when making decisions.

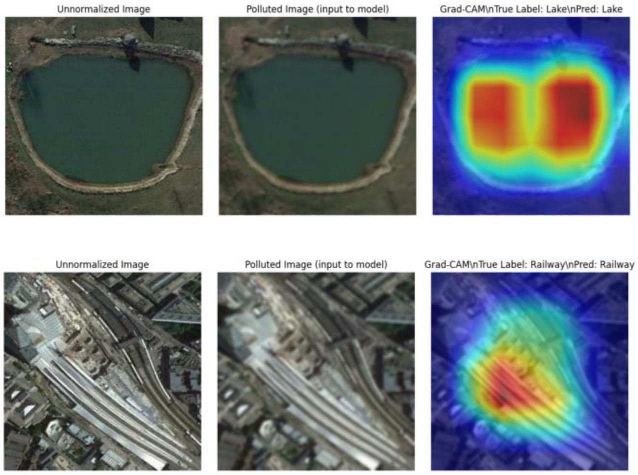
6.1 Comparison of Attention Maps

Table 2: Comparison of attention map features between ResNet and EfficientNet

Feature	ResNet	EfficientNet
Scaling Strategy	Fixed depth with residual blocks	Compound scaling (depth, width, resolution)
Receptive Field	Larger, but less controlled	Balanced and efficiently scaled
Attention Spread	More diffused across multiple regions	More focused on key areas
Localization	Rough and broader activation	Sharper and more precise localization
Interpretability	Highlights general object shape	Emphasizes most discriminative part

Following this summary, we observe that ResNet's fixed-depth residual architecture produces broader, more diffuse activation maps: its larger but uncontrolled receptive fields capture extensive context at the cost of precision. In contrast, EfficientNet's compound scaling yields well-balanced receptive fields, focusing attention sharply on the most discriminative regions. Consequently, EfficientNet's Grad-CAM heatmaps localize key semantic features (e.g., vehicle edges, rooflines) more precisely, which aligns with its higher classification accuracy. ResNet, while offering a more holistic view of the scene, may dilute class-specific signals, explaining occasional misclassifications in cluttered backgrounds.

Figure 5 - 7: Attention maps of EfficientNet


Unnormalized Image
Polluted Image (input to model)
Grad-CAM\nTrue Label: Lake\nPred: Lake
Unnormalized Image
Polluted Image (input to model)
Grad-CAM\nTrue Label: Railway\nPred: Railway

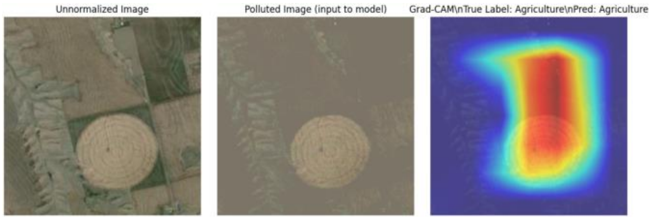
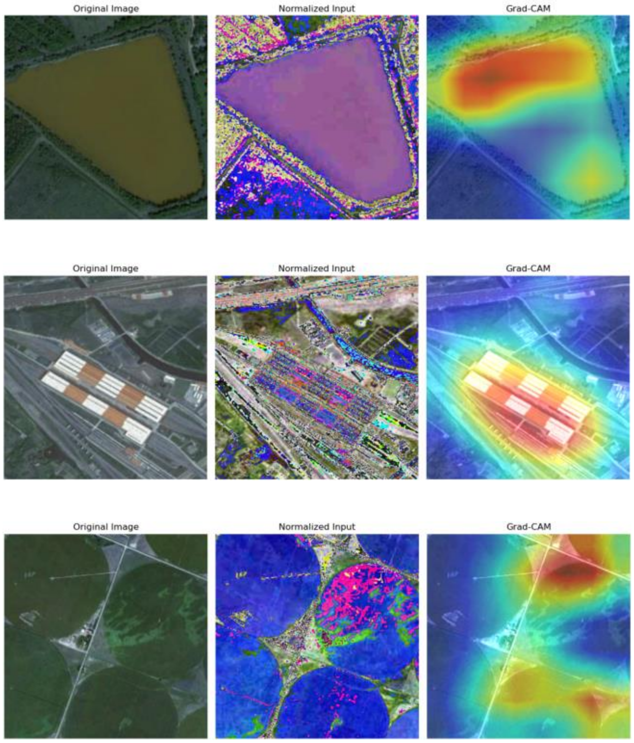

Unnormalized Image
Polluted Image (input to model)
Grad-CAM\nTrue Label: Agriculture\nPred: Agriculture

Figure 8 – 10: Attention maps of ResNet



6.2 Misclassification Insight

Grad-CAM was particularly helpful in understanding why misclassifications occurred. In cases where "commercial" was misclassified as "industrial," the heatmaps showed attention on generic rooftops or vehicles, which are present in both classes. Similarly, "lake" and "river" confusions occurred when the model fixated on open water surfaces without capturing shoreline geometry.

These observations suggest that even high-performing models can suffer from semantic ambiguity in aerial scenes, especially when distinguishing between functionally or visually similar landscapes.

6.3 Cross-model Attention Consistency

By comparing Grad-CAM outputs across models for the same image, we found that EfficientNet's attention maps were not only more localized but also more consistent across images in the same class. ResNet18 showed more variability in its focus, sometimes shifting between class-relevant and background regions. This consistency in EfficientNet may contribute to its superior classification performance and lower misclassification rates.

6.4 Usefulness for Trust and Debugging

From a practical standpoint, Grad-CAM visualizations serve two important roles:

a) Debugging Tool – By examining attention regions, we identified several model failure cases that were not easily detectable through metrics alone. This guided our error analysis and prompted discussions around dataset augmentation and class similarity.

b) Trust Enhancement – For end-users or domain experts (e.g., urban planners or environmental researchers), seeing where a model is "looking" helps justify its predictions, making it easier to adopt in decision-making pipelines.

6.5 Limitations and Future Directions

While Grad-CAM is useful for coarse localization, it lacks fine-grained precision and may fail in highly cluttered scenes. It also requires gradient access, making it unsuitable for certain deployment environments. In future work, we plan to explore complementary techniques such as:

a) LayerCAM for sharper focus on lower layers,

b) Score-CAM for gradient-free explanations,

c) Occlusion Sensitivity to test region importance via perturbation.

Integrating these methods with semi-supervised learning or attention-guided training could further boost both performance and transparency.

7. Discussion

Our comparative study reveals several important insights into the strengths and limitations of traditional feature-engineering pipelines versus modern deep learning approaches for aerial scene classification.

7.1 Trade-off Between Handcrafted and Learned Features

Traditional methods such as LBP + kNN and SIFT + SVM + SPM remain attractive for their simplicity, low computational requirements, and interpretability. LBP's texture histograms and SIFT's scale- and rotation-invariant descriptors can be extracted quickly—even on CPU-only systems—and their decision boundaries are easy to inspect. However, these handcrafted features inherently capture only local patterns and often fail to model global context, which limits their ability to distinguish between complex classes that share similar low-level cues (e.g., "industrial" vs. "commercial" or "lake" vs. "river"). This is reflected in their moderate accuracies (68.4% and 73.6%, respectively) and pronounced performance drops under perturbations like Gaussian blur or occlusion (e.g., a 37.2% F1 decline for SIFT + SPM).

In contrast, deep CNNs learn hierarchical representations that encode both local textures and high-level semantics. ResNet18 achieved a substantial accuracy boost (90.2%), while EfficientNet-B0 further improved to 93.1%, benefiting from compound scaling of depth, width, and input resolution. These end-to-end models generalize far better on the clean SkyView images, especially for visually complex or heterogeneous classes such as "industrial/commercial" or "residential."

7.2 Robustness and Architectural Resilience

While deep models excel on uncorrupted data, our perturbation experiments show that not all architectures are equally robust. ResNet18's residual skip connections help preserve feature integrity under noise and occlusion—its accuracy degrades by only 4.9%—whereas EfficientNet-B0, despite its higher clean-data accuracy, suffers a 10.7% drop under the same conditions. This suggests a trade-off: EfficientNet's optimized parameter efficiency may come at the expense of tolerance to structural distortions. Practitioners deploying in real-world scenarios—where airborne sensors frequently capture motion blur, atmospheric haze, or partial occlusion—should carefully evaluate the robustness of their chosen backbone and consider augmentations or defense mechanisms (e.g., adversarial training, occlusion sensitivity testing).

7.3 Interpretability Through Grad-CAM

The adoption of Grad-CAM provided critical transparency into the CNN decision process. EfficientNet-B0's heatmaps were consistently sharper and more semantically aligned—highlighting roads, rooftops, or water bodies—where ResNet18 sometimes produced broader, less discriminative activations. These visualizations helped pinpoint failure modes, such as when models focused on background clutter rather than key features, leading to "lake" vs. "river" mix-ups. Embedding such explainability tools into operational pipelines can aid domain experts (e.g., urban planners, environmental scientists) in validating predictions, diagnosing model biases, and building trust in automated systems.

7.4 Computational Considerations

From a resource standpoint, traditional pipelines train in minutes on commodity hardware and may be preferable for rapid prototyping or edge deployments with strict latency constraints. Deep models, however, require GPU acceleration and tens of minutes per training run. Yet, their inference times remain practical and can be further optimized via model pruning or quantization. The decision to adopt deep learning should therefore account for available compute resources, real-time requirements, and the cost of misclassifications in the target application.

7.5 Limitations and Outlook

Our unified evaluation on the SkyView dataset provides a clear baseline, but several questions remain. First, cross-dataset generalization—testing on imagery from different sensors or geographic regions—could uncover domain shift vulnerabilities. Second, low-data regimes and highly imbalanced class distributions, common in remote sensing, were not explored here; semi-supervised or self-supervised pretraining may help. Finally, while Grad-CAM offers coarse localization, finer-grained interpretability methods (e.g., LayerCAM, Score-CAM, or occlusion sensitivity maps) merit investigation.

In summary, while deep CNNs markedly outperform traditional methods in accuracy and semantic understanding, their robustness and computational demands present trade-offs that must be carefully managed. Integrating interpretability tools and robustness evaluations is crucial for deploying trustworthy aerial scene classifiers in operational environments.

8. Conclusion

In this study, we conducted a comprehensive evaluation of four aerial scene classification pipelines—LBP + kNN, SIFT + SVM + SPM, ResNet18 + Grad-CAM, and EfficientNet-B0 + Grad-CAM—on the balanced SkyView dataset (12,000 images, 15 classes). Our results demonstrate that:

a) Deep learning methods far outperform traditional feature-engineering approaches, with ResNet18 achieving 90.2% accuracy and EfficientNet-B0 reaching 93.1%, compared to 73.6% for SIFT + SPM and 68.4% for LBP + kNN.

b) Robustness varies by architecture: under occlusion and noise, ResNet18's accuracy drops only 4.9% (versus 10.7% for EfficientNet-B0), while traditional methods suffer severe degradation under blur (SIFT + SPM's F1-score falls 37.2%).

c) Interpretability via Grad-CAM reveals that EfficientNet-B0 focuses more sharply on semantically relevant regions—such as roads, rooftops, and water bodies—than ResNet18, aiding in diagnosing misclassifications between visually similar categories.

d) Computational efficiency considerations show that lightweight pipelines train in minutes on CPU, whereas deep models leverage GPU acceleration to train within tens of minutes, underscoring the trade-off between accuracy gains and resource costs.

8.1 Limitations

a) Grad-CAM provides coarse, sometimes inconsistent localization and may not fully capture fine-grained decision cues.

b) Deep models require substantial labeled data and may underperform in low-data regimes.

c) Our experiments were confined to a single dataset; cross-dataset generalization remains untested.

d) EfficientNet-B0's sensitivity to distortions suggests a need for more robust training strategies.

8.2 Future Work

a) Advanced Pretrained Architectures
Explore vision transformers (ViT), Swin Transformer, and other self-attention–based models to capture long-range dependencies and enhance domain adaptation.

b) Self-Supervised and Semi-Supervised Pretraining
Leverage contrastive learning frameworks (e.g., SimCLR, MoCo) and techniques like Pseudo-Label or FixMatch to reduce reliance on labeled data and improve feature representations for underrepresented classes.

c) Dynamic Data Augmentation
Employ reinforcement learning or evolutionary algorithms (AutoAugment, RandAugment) to automatically discover optimal augmentation policies and adapt augmentation intensity per class or scenario.

d) Enhanced Explainability Techniques
Incorporate methods such as Score-CAM, LayerCAM, and Occlusion Sensitivity for finer-grained, gradient-free visual explanations of deep models.

e) Hybrid and Multi-Modal Fusion
Combine handcrafted descriptors (LBP, SIFT) with deep features, and integrate optical imagery with LiDAR or hyperspectral data to enrich scene representation.

f) Cross-Dataset and Transfer Evaluation
Validate model robustness and generalization on diverse aerial datasets with varying resolutions, class definitions, and acquisition conditions.

By pursuing these directions, future aerial scene classification systems can achieve higher accuracy, greater robustness, and deeper interpretability—ultimately enabling more reliable land-use monitoring, rapid disaster assessment, and intelligent urban planning in real-world remote sensing applications.

9. Appendix

Command Line Examples:

```
python classify.py --model resnet --data ./SkyView
python extract_features.py --method sift --output bow.pkl
```

Grad-CAM was generated using: https://github.com/jacobgil/pytorch-grad-cam

Label Distribution:
Each class contains 800 images. No class imbalance in base dataset. Future work includes evaluating under synthetic imbalance.

10. References

[1] G. Cheng, J. Han, and X. Lu, "Remote Sensing Image Scene Classification: Benchmark and State of the Art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017. Link: https://arxiv.org/abs/1703.00121

[2] Y. Wang *et al.*, "A Survey on Deep Learning-Driven Remote Sensing Image Scene Classification," *Appl. Sci.*, vol. 9, no. 10, Art. 2110, Oct. 2019. Link: https://www.mdpi.com/2076-3417/9/10/2110

[3] S. Munir *et al.*, "Lightweight Deep Learning Models for Aerial Scene Classification: A Survey," *Pattern Recognit. Lett.*, vol. 183, pp. 1–15, Feb. 2024. Link: https://www.sciencedirect.com/science/article/pii/S0952197624020189 ScienceDirect

[4] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Trans. PAMI*, vol. 24, no. 7, pp. 971–987, 2002. Link: https://www.cse.msu.edu/~rossarun/BiometricsTextBook/Papers/Face/Ojala_LBP_PAMI02.pdf

[5] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004. Link: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[6] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE CVPR*, 2006, pp. 2169–2178. Link: https://inc.ucsd.edu/mplab/users/marni/Igert/Lazebnik_06.pdf

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE CVPR*, 2016, pp. 770–778. Link: https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[8] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *Proc. ICML*, 2019, pp. 6105–6114. Link: https://arxiv.org/abs/1905.11946

[9] R. Selvaraju *et al.*, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *Proc. ICCV*, 2017, pp. 618–626. Link: https://openaccess.thecvf.com/content_ICCV_2017/papers/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.pdf

[10] A. Aleissaee et al., "Transformers in Remote Sensing: A Survey," arXiv:2209.01206, 2022. Link: https://arxiv.org/abs/2209.01206

[11] X. Wang and A. Tien, "Remote Sensing Scene Classification with Masked Image Modeling (MIM)," arXiv:2302.14256, 2023. Link: https://arxiv.org/abs/2302.14256

[12] P. Gómez and G. Meoni, "MSMatch: Semi-Supervised Multispectral Scene Classification with Few Labels," arXiv:2103.10368, 2021. Link: https://arxiv.org/abs/2103.10368

[13] J. Zhu *et al.*, "MVP: Meta Visual Prompt Tuning for Few-Shot Remote Sensing Image Scene Classification," arXiv:2309.09276, 2023. Link: https://arxiv.org/abs/2309.09276

[14] C. Tao *et al.*, "Remote Sensing Image Scene Classification with Self-Supervised Paradigm under Limited Labeled Samples," arXiv:2010.00882, 2020. Link: https://arxiv.org/abs/2010.00882