Comparative Study of Traditional Machine Learning and Deep Learning Methods for Aerial Scene Classification Using the SkyView Dataset

1st Caizhuangzhou Cui
School of CSE
University of New South Wales
Sydney, Australia
z5509061@ad.unsw.edu.au

2nd Haomiao Zhao
School of CSE
University of New South Wales
Sydney, Australia
z5521569@ad.unsw.edu.au

3rd Jianchao Li
School of CSE
University of New South Wales
Sydney, Australia
z5319265@ad.unsw.edu.au

4th Jingqi Yang
School of CSE
University of New South Wales
Sydney, Australia
z5578536@ad.unsw.edu.au

5th Zihan Chen
School of CSE
University of New South Wales
Sydney, Australia
z5527498@ad.unsw.edu.au

Abstract—**Aerial scene classification stands important for urban planning together with environmental monitoring as well as disaster response applications. The research investigates three approaches for classifying high-definition aerial images contained within the SkyView Aerial Landscape Dataset that has 15 distinct scenes across 12,000 aerial pictures. We developed an automatic processing system that used SVM and KNN classifiers with SIFT and LBP features while testing it with ResNet-18 and EfficientNet-B0 deep learning algorithms through transfer learning. Model robustness gained improvement through the application of data augmentation methods that included rotation variances combined with cropping approaches and brightness adjustment modifications. Grad-CAM was integrated to visualize model attention as an interpretability improvement method. Deep learning approaches prove better than ordinary methods in accuracy as well as generalization through testing. Additionally Grad-CAM generates informative visual models. The study demonstrates how competing factors between accuracy performance brackets against efficiency speed and explainable interpretability operate in scene classification systems.**

Keywords—Aerial Scene Classification, Deep Learning,ResNet-18, EfficientNet-B0, Traditional Machine Learning, SIFT, Grad-CAM, Data Augmentation, Transfer Learning, Explainable AI, Large Language Model

I. INTRODUCTION

The identification of aerial imagery scenes belongs to essential functions in remote sensing since it enables many different applications. The recognition framework supports urban planning by monitoring land use and infrastructure while performing environmental surveillance by tracking deforestation or water body changes and being used during disaster response to identify fire or flood zones. The goal of this project uses the SkyView Aerial Landscape Dataset available through Kaggle to resolve this problem with its 800 high-resolution images distributed across 15 equally balanced categories.

The high-quality balanced dataset presents a complicated challenge for the classification problem despite its excellence. Some pairs of categories share similar visual characteristics (desert and mountain as well as sea and river) making it difficult for classifiers to distinguish them properly. The model generalization becomes impaired because intra-class variations emerge from factors such as lighting and viewpoint differences and weather conditions. Deep learning models need extensive large-scale data for peak performance but this dataset consists of only moderate data size. The implementation of high accuracy requires explainable AI techniques alongside it to interpret a model's behavior.

The solution to these problems requires the implementation of three different methods which we evaluate. This study adopts the combination of manually constructed features SIFT and LBP alongside SVM and KNN classifier methods to establish its first technique. Two other methods use convolutional neural networks (CNNs) with ResNet-18 and EfficientNet-B0 implementing transfer learning for their optimization. The team applies different data augmentation methods for better generalization and uses Grad-CAM visualization to display model attention. The study evaluates classifier performance alongside an interpretation of algorithms for real-world implementation of the solutions.

II. LITERATURE REVIEW

A. Traditional Methods

The widespread adoption of deep learning as a popular method coincided with visual identification being performed with handcrafted features together with conventional machine learning techniques. The key points and local features detected by SIFT (Scale-Invariant Feature Transform) maintain stability during rotational changes and scaling transformations [1]. Local Binary Patterns (LBP) serves as one of the most popular methods which detects texture patterns through basic implementations [2]. The classification process requires us to handle obtained features through SVM (Support Vector Machine) although this model works to identify the optimal class separation point while operating in high-dimensional spaces. KNN (K-Nearest Neighbors) stands as a typical classification approach that establishes predictions through neighboring points vote on class labels. The implementation of these methods demonstrates high speed while being understandable to explain. The methods prove ineffective when dealing with images of complexity or when class separation relies on global patterns. A classifier depends on high-quality features for effective operation.

B. Deep Learning

Image classification underwent a significant transformation through Convolutional Neural Networks because these models learn features straight from the data. Multiple layers with filters within them allow detection of local patterns that become more complex and abstract in subsequent layers. The CNN family contains ResNet [3] as well as EfficientNet [4] and SENet [5]. Our project employs the models ResNet-18 together with EfficientNet-B0 because they provide reasonable training capability across the dimensions of our medium-sized dataset. The architecture of ResNet-18 incorporates residual blocks which ensure easy gradient flow and prevent gradient vanishing issues. Such network architecture proves beneficial for deep networks. The EfficientNet-B0 model combines fast performance with efficient scaling among depth and width parameters along with input image dimensions. The network produces successful performance while working with fewer parameters. Our dataset receives fine-tuning from pre-trained model versions which ImageNet provides. Transfer learning represents a useful technique that works when available images reach their limit.

C. Class Imbalance and Long-Tail Learning

Data collected from actual systems contains classes whose occurrences exceed those of others in the data set. Such distribution patterns between classes are officially known as class imbalance or long-tail distribution. The increased number of one type of sample can make the model focus excessively on these samples and disregard the smaller ones. The problem of class imbalance affects scene classification vignettes particularly within real satellite or drone imagery. Zhang et al. [7] provide several methods to address such problem in their research paper. Oversampling rare classes is one solution combined with giving the loss function more weight to minority classes and creating specialized augmented training data primarily for minority classes. We utilized balanced datasets by assigning 800 images to each class therefore we omitted application of these methods for now. These approaches have potential value when performing simulations of actual problems.

D. Explainable AI and Grad-CAM

The superior accuracy achieved by deep learning models comes with a core limitation of being difficult or impossible to interpret because of their opaque nature. The inadequate visibility about prediction processes presents a major restraint when working with applications that require strict privacy such as disaster management and environmental monitoring.

XAI (Explainable AI) functions to solve this issue by providing explanations about model prediction processes. Among XAI methods Grad-CAM (Gradient-weighted Class Activation Mapping) by its name stands out as it creates heatmaps to display the regions that shape predictions the most [6]. This technique leverages the gradients of the output with respect to convolutional feature maps to localize class-discriminative regions.

Our research uses Grad-CAM for analyzing correctly and incorrectly classified images within various models. The analysis lets us verify how well models spot important semantic features and how much they depend on distractive features thus helping understand the model's behavior as well as possible failure modes.

E. LLMs for Post-hoc Validation and Hybrid Modeling

The scientific community now explores the utility of Large Language Models (LLMs) including GPT-4 to handle vision-related missions through their effective semantic processing abilities [8]. LLMs possess superior semantic reasoning ability to deliver conceptual interpretations from analyzing image meaning with accompanying text compared to traditional CNN pixel-based recognition systems [9].

GPT-4o participated in the project to reevaluate samples that EfficientNet-B0 had wrongly identified. The model uses LLM-based research methodology during post-hoc validation to explain images for predicting appropriate labels. The experimental results indicate a potential approach to incorporate human-like semantic processing into visual analysis systems even though low overall accuracy existed because of limited input and visually similar class samples.

An integration of CNN visual perception together with LLM reasoning-based models shows potential to establish better interpretability and eliminate system blind spots while facilitating human-assisted operations. Future extensions could involve LLMs receiving instructions to generate explanations for CNN predictions while they might provide feedback about the training process and operational stages [10].

Manufacturer-made image categorization methods including SIFT and LBP evolved into deep learning systems like ResNet and EfficientNet that automatically discover sophisticated image patterns present in data. Deep models perform highly in accuracy yet face two important difficulties including class imbalance and lack of model interpretability. Model interpretability and fairness problems can be addressed through data augmentation and Grad-CAM together with re-weighting techniques which offer explanations about the decision-making processes of AI systems. Large language models (LLMs) created a new pathway for post-hoc validation and semantic reasoning because they present a hybrid method to enhance the comprehension and improvement of AI systems. The developed foundations will guide our experimental study about model performance and visual behavior while exploring real-world aerial scene classification deployment potential.

III. Methods

Three methodologies are implemented and tested for the classification process. Traditional machine learning features extraction combined with classifier forms the first method. Our research has two deep learning methods based on ResNet-18 and EfficientNet-B0. The same dataset split serves both training and testing phases for all methods while matching metrics determine their evaluation results.

A. Data Preprocessing and Augmentation

The research employs the SkyView dataset with fifteen scenes divided into 800 images for each class. Each class was divided into training and testing sections where 80% of images served for training (640 images) and 20% were used for testing (160 images). The training set contains 9,600 images together with 2,400 images in the test set. Training set data augmentation includes five methods: Horizontal Flips for mirror-like area views and Rotations (±15°) ensuring model stability and Cropped images with resizing for partial views and Brightness modifications for various light conditions and adding Gaussian Blur to reduce sensitivity to image quality or blurriness. Every original image undergoes five augmentations so the data set grows to 6 copies of each image. A total of 57,600 images can be found in the ultimate training collection which contains 6 variations of each of the 9,600 images. The test dataset remains unchanged without any augmentation process.

B. Traditional Machine Learning Method

We apply SIFT for detecting key points alongside descriptors of local patterns alongside LBP which produces histograms from local texture patterns. Secondly we employ classifiers for training which include SVM with RBF kernel as well as KNN at k values of 3, 5, and 7. The methods operate at speed while requiring no GPU to conduct training operations. Models function based on the quality of features selected. Confusion matrix provides insights into which classes the model frequently mistakes while performing its task thus enabling better comprehension of model operations.

C. Deep Learning Method 1: ResNet-18

The ResNet-18 model available in PyTorch framework receives ImageNet weights for its operation. Our model requires a 15-class configurable output layer instead of its original configuration. During training the system used the following parameters: Input dimension at 224×224 and a 32-batch size with learning rate set to 1e-4 while employing Adam optimizer and Cross Entropy loss over 25 training epochs. Data preprocessing followed the previous methods. During testing we apply Grad-CAM to produce heatmaps for visualizing testing images. The analysis of model learning progresses from both proper and improper predictions.

D. Deep Learning Method 2: EfficientNet-B0

Our research includes training of the EfficientNet-B0 model which represents a smaller and efficient architectural version. We employ a pretrained version which we modify by updating the most immediate layers only. The other training parameters correspond exactly to ResNet design. EfficientNet needs lower memory and operates at a higher speed yet demonstrates slightly less accurate predictions than ResNet. The analysis of attention zones between ResNet and EfficientNet employs Grad-CAM methodology for a second time.

E. Evaluation Metrics

Our evaluation of model performance includes the three established metrics which consist of Precision to measure correct predictions and Recall to measure correct real sample findings along with F1-score as their mean value. These tools comprise the Confusion Matrix which indicates difficult classification points and the Sample Visualization tool provides correct and incorrect predictions and the Grad-CAM Heatmaps for each case.

F. LLM-Based Post-hoc Verification

The classification system incorporates GPT-4o as a large language model to evaluate post-hoc the misclassified output instances of EfficientNet-B0. The tool functions exclusively to help determine if model mistakes are justifiable or reducible.

We provided 43 images which EfficientNet mistakenly classified to GPT-4o for semantic image re-assessment. The LLM received both the images and their designated classifications. AI systems composed like human intelligence need to demonstrate their ability to confirm when vision models produce inaccurate outputs.

The evaluation metrics such as accuracy and precision measured the LLM's performance just like the model assessment while a confusion matrix displayed its judgment patterns. This analysis provided important knowledge about semantic label mix-ups in different categories (“Highway” and “Railway” and “Lake” and “River”) while demonstrating a low overall classification accuracy rate of ~9.3%.

IV. Experimental Results

We compare the performance of three different methods for 15-class aerial image classification:

1.Traditional machine learning (SIFT/LBP + SVM)
2. Deep learning using ResNet-18
3. Deep learning using EfficientNet-B0

A. Overall Accuracy Comparison

Table 1

Model	Feature Type	Accuracy (%)	Notes
SVM (SIFT+LBP)	Handcrafted Features	0.7178	High speed, lower generalization
ResNet-18	Deep Features	0.9171	Better than SVM
EfficientNet-B0	Deep Features	0.9850	Best performance overall

Note: All models are evaluated on the same 2,400-image test set. Accuracy is reported as top-1 accuracy.

B. Classification Report

Below are selected class-level results comparing model precision, recall, and F1-score:

Table 2

Model	Precision	Recall	F1-score
SVM (SIFT+LBP)	0.7184	0.7177	0.7167

ResNet-18	0.9188	0.9171	0.9171
EfficientNet-B0	0.9870	0.9851	0,9850

C. Confusion Matrix

To examine which categories are confused most often, we plot the confusion matrix for all models.

Figure 1. Confusion Matrix – SVM

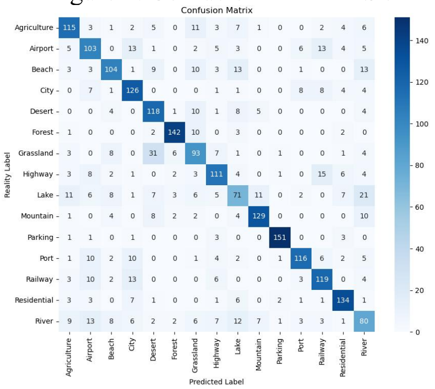


Figure 2. Confusion Matrix – ResNet-18

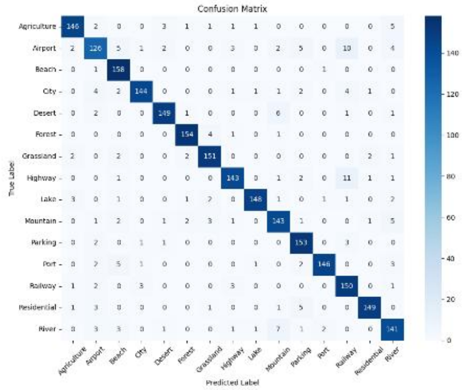


Figure 3. Confusion Matrix – EfficientNet-B0

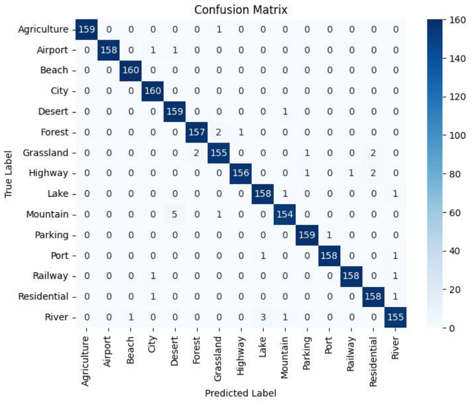
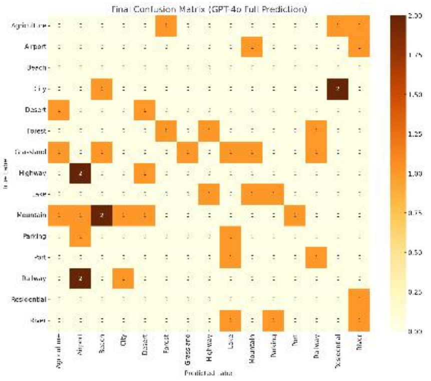


Figure 4. Confusion Matrix – GPT-4o



D. Training Curves (DL Models Only)

Figure 5. ResNet-18 Accuracy & Loss vs Epoch

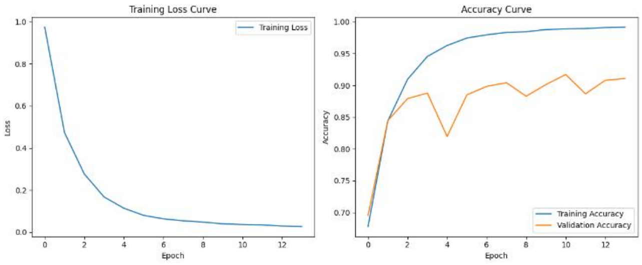
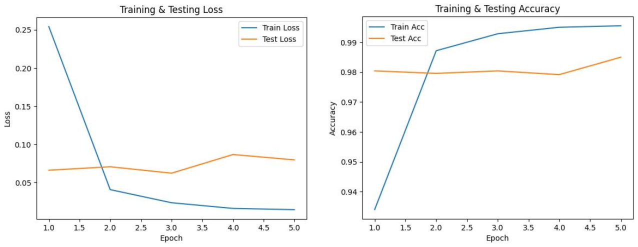


Figure 6. EfficientNet-B0 Accuracy & Loss vs Epoch



E. Grad-CAM(Left:ResNet-18, Right:EfficientNet-B0)

Figure 7. Parking

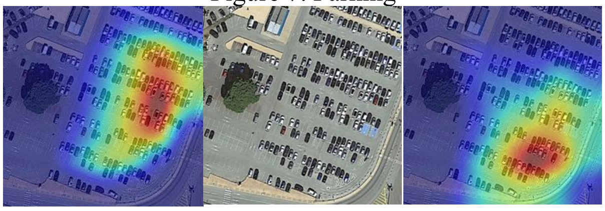


Figure 8. Highway

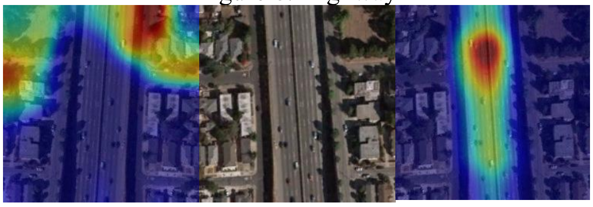


Figure 9. Airport

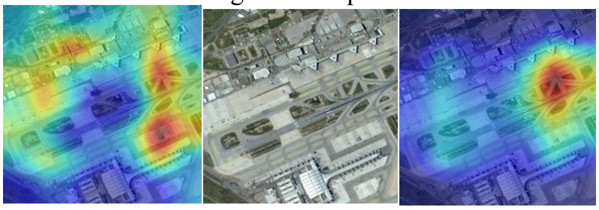


Figure 10. Port

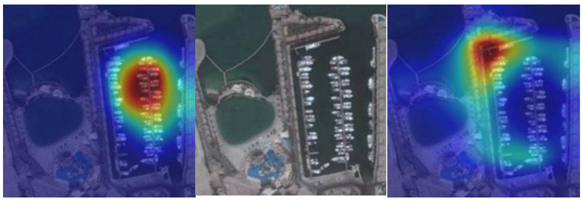
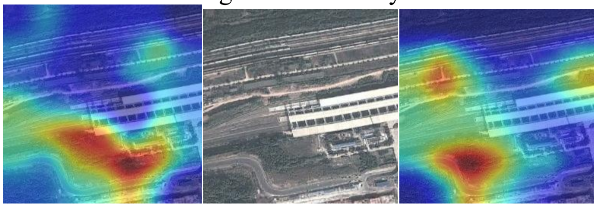
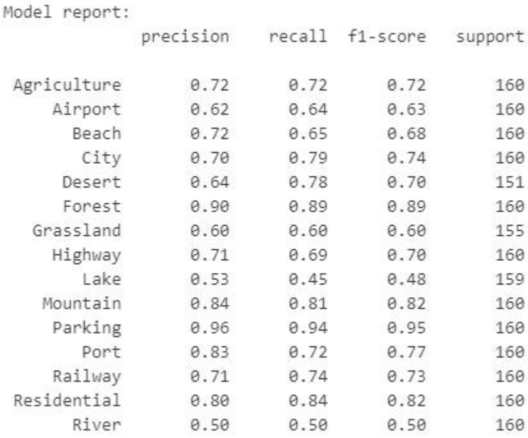


Figure 11. Railway



F. Model Report

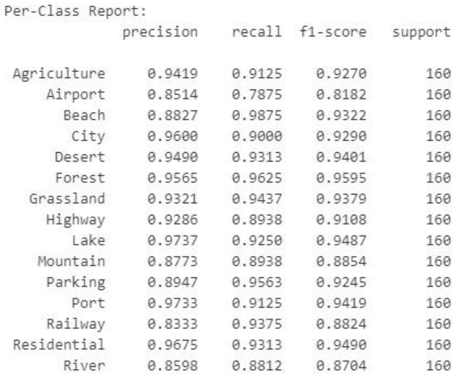
Figure 12. Model Report —SVM



```
Model report:
              precision    recall  f1-score   support

 Agriculture       0.72      0.72      0.72       160
     Airport       0.62      0.64      0.63       160
       Beach       0.72      0.65      0.68       160
        City       0.70      0.79      0.74       160
      Desert       0.64      0.78      0.70       151
      Forest       0.90      0.89      0.89       160
   Grassland       0.60      0.60      0.60       155
     Highway       0.71      0.69      0.70       160
        Lake       0.53      0.45      0.48       159
    Mountain       0.84      0.81      0.82       160
     Parking       0.96      0.94      0.95       160
        Port       0.83      0.72      0.77       160
     Railway       0.71      0.74      0.73       160
 Residential       0.80      0.84      0.82       160
       River       0.50      0.50      0.50       160
```

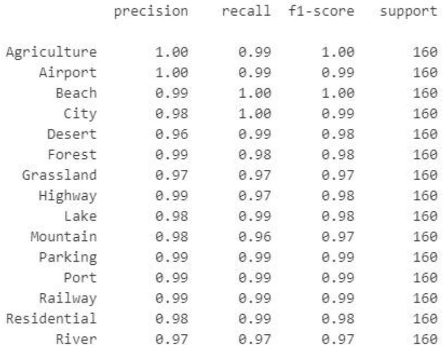
Figure 13. Model Report —ResNet-18



```
Per-Class Report:
              precision    recall  f1-score   support

 Agriculture     0.9419    0.9125    0.9270       160
     Airport     0.8514    0.7875    0.8182       160
       Beach     0.8827    0.9875    0.9322       160
        City     0.9600    0.9000    0.9290       160
      Desert     0.9490    0.9313    0.9401       160
      Forest     0.9565    0.9625    0.9595       160
   Grassland     0.9321    0.9437    0.9379       160
     Highway     0.9286    0.8938    0.9108       160
        Lake     0.9737    0.9250    0.9487       160
    Mountain     0.8773    0.8938    0.8854       160
     Parking     0.8947    0.9563    0.9245       160
        Port     0.9733    0.9125    0.9419       160
     Railway     0.8333    0.9375    0.8824       160
 Residential     0.9675    0.9313    0.9490       160
       River     0.8598    0.8812    0.8704       160
```

Figure 14. Model Report —EfficientNet-B0



```
              precision    recall  f1-score   support

 Agriculture       1.00      0.99      1.00       160
     Airport       1.00      0.99      0.99       160
       Beach       0.99      1.00      1.00       160
        City       0.98      1.00      0.99       160
      Desert       0.96      0.99      0.98       160
      Forest       0.99      0.98      0.98       160
   Grassland       0.97      0.97      0.97       160
     Highway       0.99      0.97      0.98       160
        Lake       0.98      0.99      0.98       160
    Mountain       0.98      0.96      0.97       160
     Parking       0.99      0.99      0.99       160
        Port       0.99      0.99      0.99       160
     Railway       0.99      0.99      0.99       160
 Residential       0.98      0.99      0.98       160
       River       0.97      0.97      0.97       160
```

V. DISCUSSION

This section performs a comprehensive investigation of three implemented methods—SVM with handcrafted features, ResNet-18, and EfficientNet-B0—for their classification performance together with their learning behaviors and interpretability and deployment aspects. Our main objective is to evaluate quantitative performance alongside discovering fundamental reasons for achievements and shortcomings in the methods.

A. Overall Performance Comparison

The classification accuracy of EfficientNet-B0 reaches 98.50% which constitutes the highest performance when compared to ResNet-18 at 91.71% and SVM at 71.78%. The F1-score performance demonstrates EfficientNet leads at 0.9850 while ResNet-18 obtains 0.9171 and SVM has 0.7167.

The compound scaling mechanism together with improved parameter efficiency make EfficientNet-B0 perform better than its counterparts. ResNet-18, while a robust CNN baseline, shows slightly lower generalization capacity. The combination of SIFT and LBP features used by the traditional SVM model fails to comprehend advanced semantics so it produces mediocre results in situations with complex elements.

B. Class-Wise Performance and Confusion Analysis

Different classification score reports show the following important trends:

SVM fails on visually similar or texture-ambiguous classes such as Lake (F1 = 0.48), River (F1 = 0.50), and Grassland (F1 = 0.60). The classification results for defined textures like Parking demonstrate excellent scores reflected by an F1 measure of 0.95.

The performance of ResNet-18 increases for Lake and Port categories while maintaining a moderate level of confusion that affects Mountain and Forest identification.

The network reaches nearly perfect accuracy in every single ImageNet category while the lowest F1-score remains above 0.97 including difficult cases such as the River and Grassland classes.

The results from confusion matrices support these evaluation findings.

SVM makes widespread misclassifications.

The ResNet model cuts down on most mistaken classifications while leaving some natural scene identifications incorrect.

The data distribution in EfficientNet leads to matrices with well-diagonal organization and very low error dispersion.

C. Training Behavior and Learning Curves

Figure 1 indicates a significant separation between training accuracy and testing accuracy because strong overfitting and poor generalization occur. In contrast:

ResNet-18 (Figure 5) shows steady convergence and manageable overfitting. The validation accuracy achieves a target range between 0.87 to 0.91 during which the training loss follows a smooth decay pattern.

EfficientNet-B0 (Figure 6) demonstrates excellent learning dynamics. The model trains rapidly while keeping training and testing accuracy at around 0.99 and 0.98 simultaneously and produces negligible levels of overfitting.

Without changing the number of epochs EfficientNet displays good training efficiency and resistance to abnormal data patterns.

D. Model Trade-offs and Deployment Considerations

Table 3

Model	Training Time	Generalization	Strengths	Limitations
SVM	Very Fast	Weak	Lightweight, interpretable	Low accuracy, poor high-level features
resnet	Moderates	Good	Strong baseline, CNN flexibility	Slight confusion in similar classes
efficientnet	Fast	Excellent	Best accuracy, efficient and interpretable	Slightly higher resource requirement

E. Interpretability Analysis via Grad-CAM

Grad-CAM analysis enabled understanding the visual reasoning processes for ResNet-18 and EfficientNet-B0 when processing five different scene types. Figures 6 through 10 present attention maps for both ResNet-18 and EfficientNet-B0 on the same visual samples.

Case Study 1: Railway

The EfficientNet-B0 model shows precise track and platform orientation (Figure 7-a) whereas ResNet-18 distributes its attention across wider non-specific areas (Figure 7-b).

Case Study 2: Port

The EfficientNet model focuses its attention on both dockside areas along with moored vessels as illustrated in Figure 8-a. The attentional network of ResNet-18 spreads its evaluation toward the surrounding bodies of water in this image (Figure 8-b).

Case Study 3: Airport

The area of focus for EfficientNet runs directly along terminal routes and focuses on central areas (Figure 9-a) but ResNet shows less distinct concentration on a wider scope (Figure 9-b).

Case Study 4: Highway

The commercial highways receive excellent precision from EfficientNet in Figure 10-a yet ResNet intersects between the highways and adjacent urban elements in Figure 10-b.

Case Study 5: Parking

EfficientNet performs accurate cluster detection of vehicles while ResNet spreads its focus over surrounding regions (Figure 11-a vs b).

Table 4

Class	EfficientNet-B0 (Focus)	ResNet-18 (Focus)
Railway	Tracks and stations	Surrounding terrain
Port	Ship docks and piers	Background water and edges
Airport	Runway and terminal core	Broader infrastructure
Highway	Center road structure	Buildings near highway
Parking	Car clusters	Nearby empty road sections

The obtained results show that EfficientNet maintains superior accuracy levels while providing interpretable visual output that focuses on important image areas which users can readily understand. The accurate outcomes achieved by ResNet-18 come with a disadvantage because it sometimes depends on visual cues outside the central focus zone which decreases user trust in its predictive judgment.

F. LLM-based Semantic Review of Misclassifications

GPT-4o performed semantic evaluation on misclassified samples of the EfficientNet-B0 model using its capability as a large language model. GPT-4o operates differently from standard CNN models since it identifies semantic categories through deep scene understanding so it performs similarly to a human reasoning agent.

The analysis featured 43 images which EfficientNet-B0 misidentified where GPT-4o independently conducted the classification task. GPT-4o achieved a low accuracy level of 9.3% on the assessment samples according to the data in Table 5. Through the confusion matrix the LLM demonstrates poor classification accuracy on visually complicated images between Highway and Railway as well as Lake and River.

Table 5

Metric Type	Accuracy	Macro Avg	Weighted Avg
Precision	9.3%	13.9%	20.7%
Recall	9.3%	8.9%	9.6%
F1-score	9.3%	9.1%	10.6%

VI. CONCLUSION

Three approaches for aerial scene classification received development and systematic evaluation as part of this project through the use of SIFT and LBP with SVM classifiers and ResNet-18 and EfficientNet-B0 deep learning methods. Our main objective involved evaluating the performance levels of these models together with assessing their internal learning patterns and their practical usage potential.

The obtained results show deep learning approaches provide superior effectiveness than traditional hand-crafted techniques in the task. The SVM model demonstrated weak ability to understand spatial and semantic characteristics in spite of its lightweight computational nature and easy interpretability by suffering major classification errors when detecting River and Lake and Forest sectors which look similar. The ResNet-18 architectural design delivered exceptional performance by providing reliable generalized results while producing patterns that researchers could interpret. The EfficientNet-B0 model surpassed all testing methods through its top-1 accuracy evaluation of 98.50% along with high F1-score performance in each classification category and required fewer training iterations and displayed narrow Grad-CAM attention patterns. The obtained research findings demonstrate that CNN model success depends heavily on designing efficient architectural frameworks.

The work also highlighted the need for models with explainable features since interpretation matters for remote sensing because transparent decisions matter in this field. The Grad-CAM visualizations show EfficientNet-B0 reaches high numerical performance and manages to focus on relevant semantic image areas like a human brain would. The error rates for ResNet occurred when the model depended on non-core image portions rather than relevant areas.

Our training and evaluation procedure generated important practical learnings. SVM operates with small training needs but faces performance restrictions when processing subtle classification requirements. The combination of performance quality alongside reduced computational requirements makes ResNet-18 suitable for deployment yet EfficientNet-B0 delivers efficient performance and resilience for deployment in real-time situations.

Large language models such as GPT-4o received consideration as part of the evaluation process for post-hoc semantic validation of incorrectly classified samples during this study. This initial test revealed how LLMs suffer from weak capabilities in visual precision yet demonstrate potential to supply semantic reasoning which helps in ambiguous situations. Explaining AI systems becomes more possible through the combination of visual attention methods together with language-based logic which creates new possibilities for future explainable AI research.

A. *Project Contributions*

This research handles a comparison of three modeling approaches between classical frameworks and contemporary theories.

The system includes every step of training from data augmentation to transfer learning to interpretability functionalities.

Grad-CAM diagnosis tools combined with confusion matrices show how the classification system functions.

Initial exploration of human-in-the-loop reasoning through LLM-assisted classification review.

The solution develops an explainable pipeline that detects model mistakes together with the reason behind such errors.

A review of model selection requires observations from both deployment restrictions and real-world use situations.

B. *Future Work*

Future developments will be achieved by following these three directions:

Enhancing recognition of minority classes becomes possible by implementing methods including class-weighted loss together with focal loss and oversampling over long-tailed class distribution scenarios.

Model robustness testing requires an evaluation of system performance while using noise and under conditions of occlusion and blur as well as adversarial perturbations to determine generalization capabilities under adverse conditions. Implement training techniques which detect noise along with methods that enhance data augmentation.

Networks with dual branches or specific attention mechanisms enable integration of DSM and NIR and metadata information from the scene to boost understanding abilities.

Researchers should examine three optimized models that include MobileNetV3 along with ShuffleNetV2 as well as neural architecture search (NAS) for lightweight architecture assessment. Districts can employ knowledge distillation to reduce the size of their large models into portable versions.

The work extends Grad-CAM analysis through LIME and SHAP combination while creating human-friendly interfaces that enable both review sessions and diagnosis of system errors.

This process involves using domain adaptation technology with self-supervised learning approaches to enable mapping of input from diverse geographical regions or sensors including satellite and drone systems.

Systems Require Full Automation for Processing Data During All Stages and Inference Through Visualization and Front-End Display At Once. The deployment of real-time applications in web or mobile interfaces requires the optimization using ONNX or TensorRT frameworks.

The integration of large language models such as GPT-4, CLIP, Flamingo with vision systems requires detailed investigation to develop their usage for semantic guidance and label verification while performing intelligent reasoning tasks. The process includes guiding LLMs to produce text explanations about CNN output results as well as supporting active learning processes.

This study provides essential groundwork to direct future research about explainable highly-performant vision systems capable of aerial scene understanding. We advance into building AI systems that merge traditional and contemporary methods combined with visualization tools and GPT-4o semantic modeling for accurate deployment in critical applications like urban planning and environmental monitoring and disaster response.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[2] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," Pattern Recognition, vol. 29, no. 1, pp. 51–59, 1996.

[3] [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2016, pp. 770–778.

[4] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.* (ICML), 2019, pp. 6105–6114.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* (CVPR), 2018, pp. 7132–7141.

[6] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.* (ICCV), 2017, pp. 618–626.

[7] Y. Zhang, Q. Wang, C. Han, X. Liu, Y. Zhang, and X. Li, "Deep long-tailed learning: A survey," IEEE Transactions on Pattern

Analysis and Machine Intelligence, vol. 45, no. 9, pp. 10795–10816, 2023.

[8] OpenAI, “GPT-4 Technical Report,” *arXiv preprint* arXiv:2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[9] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” *Proc. Int. Conf. Mach. Learn.* (ICML), 2021.

[10] J.-B. Alayrac et al., “Flamingo: a Visual Language Model for Few-Shot Learning,” *arXiv preprint* arXiv:2204.14198, 2022. [Online]. Available: https://arxiv.org/abs/2204.14198

[11] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8024–8035.

[12] TorchVision Contributors, *TorchVision: Models, Datasets and Transforms for Computer Vision*, ver. 0.17, 2025. [Online]. Available: https://github.com/pytorch/vision (accessed Apr. 23 2025).

[13] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] W. McKinney, “Data Structures for Statistical Computing in Python,” in *Proc. 9th Python in Science Conf.* (SciPy), 2010, pp. 51–56.

[16] J. D. Hunter, “Matplotlib: A 2-D Graphics Environment,*Computing inScience & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[17] M. Waskom, “Seaborn: Statistical Data Visualization,” *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[18] C. da Costa-Luis, *tqdm: A Fast, Extensible Progress Meter for Python and CLI*, ver. 4.66.3, 2023. [Online]. Available: https://github.com/tqdm/tqdm (accessed Apr. 23 2025).