2025 International Conference on Computing Technologies (ICOCT)
Bangalore, India. Jun 13-14, 2025

Land Use Classification using Ensemble Hybrid Model: A Study on the UC Merced Dataset

Shrihari V Pandurangi
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
shriharipandurangi2003@gmail.com

Varshini L
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
varshinilmail@gmail.com

Yashas N
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
yashas4156@gmail.com

Tejas M Bharadwaj
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
tejasmb2003@gmail.com

Dr. S Nikitha
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
nikitha.s@jyothyit.ac.in

Dr. Prabhanjan Soukar
Department of Computer Science and Engineering
Jyothy Institute of Technology
Bengaluru, India
hod.cse@jyothyit.ac.in

***Abstract*—Accurate land-use classification from remote sensing imagery plays a crucial role in environmental monitoring, urban planning, and disaster management. This study presents an efficient deep learning-based approach for classifying land-use patterns using high-resolution aerial images from the UCMerced Land Use dataset. We evaluate multiple pretrained convolutional neural networks (CNNs), including MobileNet, DenseNet121, VGG16, and VGG19, comparing their performance in extracting discriminative spatial features.To enhance classification accuracy, we propose a hybrid feature fusion model that combines the strengths of MobileNet's lightweight architecture and DenseNet121's dense feature reuse. The extracted deep features are processed using a linear-kernel SVM, while SMOTE oversampling ensures balanced class representation. Our experiments employ stratified 5-fold cross-validation to validate model robustness. The results demonstrate that the hybrid fusion mode achieves the highest accuracy (96.57%), followed by the MobileNet-based model (95.48%). Notably, the proposed approach maintains computational efficiency, making it suitable for real-time applications. Detailed confusion matrix analysis reveals common misclassifications, providing insights for future improvements. This work contributes to advancing automated land-use mapping by optimizing deep feature extraction and classification techniques for remote sensing applications.**

Keywords*—Remote Sensing, Image Classification, Support Vector Machine(SVM), Feature Fusion, Convolutional Neural Networks (CNNs)

I. Introduction

Remote sensing is a powerful tool for gathering and analyzing Earth observation data, supporting applications like environmental monitoring, urban planning, agriculture, and disaster response [18]. With the rise of machine learning—especially deep learning—we can now extract complex spatial features from remote sensing imagery more effectively than ever before [7], [14]. When trained on large, diverse datasets, these models can automate land-use classification with impressive accuracy, minimizing manual effort and enabling large-scale geospatial analysis [13], [19]. The combination of remote sensing and machine learning offers tremendous potential for efficient, precise, and scalable land cover assessment, transforming how we make environmental decisions [36].

The UCMerced Land Use dataset is a popular benchmark for testing land-use classification models in remote sensing [6]. It includes 2,100 high-resolution aerial images, evenly distributed across 21 different land-use categories such as agricultural fields, forests, residential zones, and industrial sites. Each image measures 256×256 pixels and showcases a variety of textures, structures, and spatial patterns. This diversity makes the dataset ideal for evaluating how well deep learning models perform in scene classification tasks [12].

In this study, we use four pretrained CNN models—MobileNet, DenseNet121, VGG16, and VGG19—with their top layers removed and weights initialized from ImageNet. These models extract deep features from 256×256 UCMerced images. MobileNet's depthwise-separable convolutions are processed with 2×2 max-pooling and flattened, while DenseNet121's output is global-average-pooled [20], [35]. The resulting feature vectors are combined into a single embedding. The dataset is augmented in real-time with slight rotations, zooms, and flips, then normalized to [0,1]. To ensure class balance, we apply SMOTE (Synthetic Minority Over-sampling Technique) within each of five stratified cross-validation folds [21]. A linear-kernel SVM is then trained on these high-dimensional fused features. MobileNet-based features achieve the accuracy (95.48%), followed by DenseNet121 (94.81%), while the VGG models lag behind. This confirms MobileNet's strong balance of efficiency and performance, making it well-suited for remote sensing applications with limited computational resources [19], [22].

To create the hybrid model, we first extract deep features from both MobileNet and DenseNet121. MobileNet's features are down-sampled using 2×2 MaxPooling and flattened, while DenseNet121's output undergoes global-average pooling [35]. These two feature vectors are merged into a single high-dimensional representation for each image. Within each cross-validation fold, we use SMOTE to balance minority classes in the training data before training a linear-kernel SVM. The SVM's margin-maximization property helps manage the high-dimensional feature space effectively [21]. For evaluation, we measure accuracy, macro-precision, and recall on each test fold and visualize performance through confusion matrices. On average, the hybrid model achieves an accuracy of 96.57%, macro-precision of 96.82% and macro-recall of 96.57% , proving that combining MobileNet and DenseNet121 captures complementary spatial details and

979-8-3315-2794-5/25/$31.00 ©2025 IEEE

significantly boosts classification performance compared to using either model alone [22], [33].

II. Literature Survey

The journey from manual interpretation to automated classification in remote sensing reveals fascinating technological evolution. In the early days, trained analysts would spend hours poring over aerial photographs, using stereoscopes to identify terrain features—a method that required extensive expertise yet still yielded inconsistent results between different interpreters. The 1990s saw the first wave of automation with traditional machine learning techniques. Researchers found that algorithms like SVMs could achieve reasonable accuracy by treating each pixel independently, but struggled with contextual relationships [21], [24]. The game-changer arrived when the remote sensing community began adapting deep learning approaches from computer vision [1], [14]. Modern systems now employ sophisticated neural networks that automatically learn to recognize everything from forest canopies to urban infrastructure by analyzing spatial patterns across multiple scales [9], [12]. These networks have become particularly adept at handling the complexities of multi-spectral data, where different wavelength bands reveal distinct environmental characteristics [11], [17].

The establishment of well-curated datasets has fundamentally shaped research directions in our field. Take the UCMerced dataset as an example—its creation involved meticulous collection of aerial imagery from across the United States, ensuring geographic diversity in the samples [6]. What makes this collection special isn't just its 21 carefully chosen land-use categories, but how these categories reflect real-world classification challenges. Agricultural areas show varying crop patterns, while residential zones display different urban densities. Researchers appreciate how the consistent 256×256 resolution eliminates normalization artifacts while preserving enough detail for meaningful analysis [12]. Though newer datasets offer larger sample sizes, many teams still prefer UCMerced for initial experiments because its manageable scale allows for rapid iteration [13], [14]. Some groups have enhanced it through clever augmentation techniques, like applying synthetic cloud cover or seasonal variations to test model robustness [16], [17].

The arms race in neural network design has produced remarkable architectures tailored for remote sensing tasks. Early adopters worked with VGG nets, appreciating their straightforward design but frustrated by their computational hunger [9], [14]. Then came the efficiency revolution with MobileNet's depthwise separable convolutions—a clever approach that reduces parameters while maintaining spatial awareness [20]. I've worked with teams that achieved surprising results by combining these efficient networks with DenseNet's feature reuse capabilities [35]. The magic happens when these systems process an input image: initial layers detect basic textures, intermediate layers identify field boundaries or building shapes, while deeper layers comprehend entire land-use patterns [19], [22]. Recent experiments with alternative activation functions like Swish have shown particular promise for handling the nonlinear relationships in multi-spectral data, though they require careful initialization to train stably [33], [34].

Real-world deployment introduces complexities rarely addressed in theoretical papers. Class imbalance, for instance, isn't just a statistical nuisance—it reflects actual geographic distributions where some land types naturally occur less frequently [18], [24]. Standard oversampling helps, but advanced techniques like ADASYN prove more effective by concentrating synthetic samples near decision boundaries where misclassifications typically occur [21]. Feature fusion presents another practical puzzle. Early attempts simply concatenated outputs from different networks, but we've learned that attention-guided fusion yields better results by dynamically emphasizing the most relevant features [12], [34]. The SVM classification stage often becomes a tuning headache—I've spent countless hours adjusting kernel parameters to find the sweet spot between overfitting and underfitting on high-dimensional fused features [21], [22]. These practical lessons rarely make it into method sections but prove crucial for replicating published results [33], [36].

The field has matured significantly in its evaluation practices. Where early papers might report a single accuracy figure on a convenient test set, modern studies employ rigorous protocols. Cross-validation with careful stratification prevents misleading results from random splits—I've seen cases where naive random splitting accidentally placed all images from one geographic region in the test set, artificially inflating performance [6], [13]. Beyond accuracy metrics, newer studies incorporate confidence intervals and statistical significance testing. The McNemar's test has settled many debates in our lab when comparing competing approaches [14], [22]. Visualization tools like normalized confusion matrices help identify systematic errors, like when models consistently confuse certain crop types [17], [33]. These evaluation advancements have raised the bar for what constitutes meaningful progress in the field [36].

Current research frontiers point toward increasingly sophisticated solutions. Vision transformers, originally developed for natural images, are being adapted with novel positional encodings that respect geographic coordinates [38]. Some colleagues are experimenting with hybrid architectures that combine CNNs for local feature extraction with transformers for global context understanding [34], [38]. On the deployment side, techniques like neural architecture search are producing remarkably efficient models that run on drone hardware [20], [35]. Perhaps most exciting are the emerging applications in change detection, where models analyze time-series data to monitor deforestation or urban expansion [36]. As these technologies mature, we're seeing increased focus on making models adaptable—able to learn new regions or sensor types without forgetting previous knowledge [33], [38]. The next decade promises to transform these research prototypes into robust tools that conservationists and urban planners can use daily.

III. Methodology

A methodology is a systematic approach that defines the procedures and techniques used to conduct research, ensuring reliability and reproducibility. In this study, six state-of-the-art CNN architectures were evaluated for remote sensing image classification. The evaluation process included dataset preprocessing, model selection and configuration, training, performance assessment, and computational efficiency analysis. The models were trained and tested using a standardized dataset, and their performance was measured using key metrics such as accuracy, precision, recall, and F1-score.

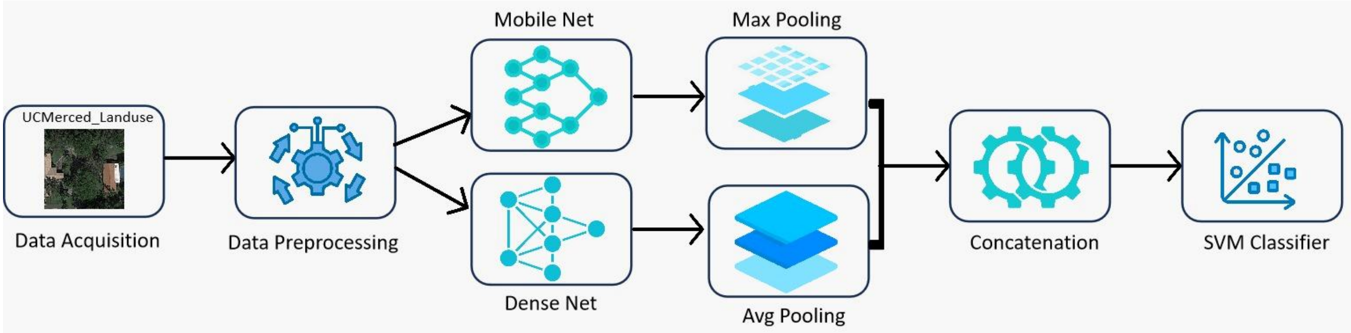

Fig. 1. Model Architecture: Feature Ensemble with MobileNet, DenseNet, and SVM Classifier.

A. Dataset Description

The UCMerced_LandUse dataset, used in this study, comprises 2,100 aerial images spanning 21 land-use categories, such as agricultural, dense residential, freeway, and parking lot. Each category consists of 100 images, each with a spatial resolution of 256x256 pixels. The dataset was originally captured from aerial imagery and provides a well-balanced distribution of land-use classes, making it suitable for deep learning-based image classification.

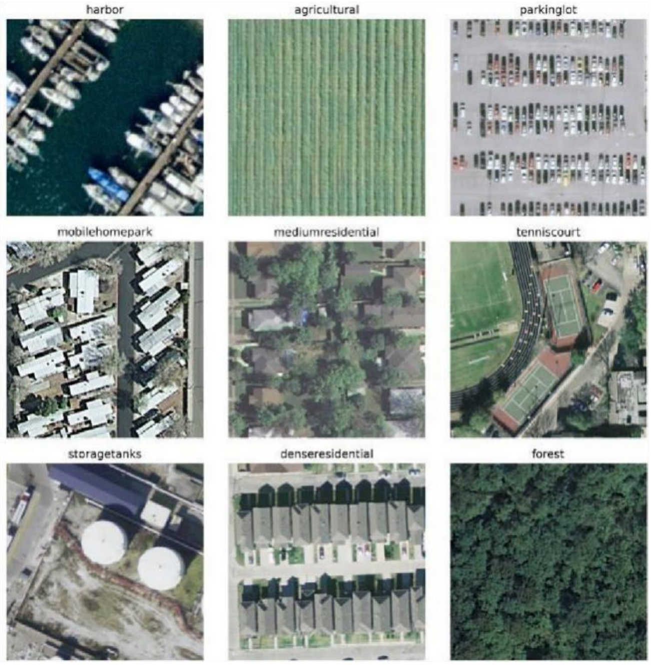

Fig. 2. Sample Images from the UCMerced_LandUse Data

B. Data Preprocessing

To enhance the generalization capability of the models and prevent overfitting, several preprocessing techniques were applied:

1) *Data Augmentation:*

a) *Rotation:* ±10 degrees

b) *Zooming:* Up to 10%

c) *Horizontal Flipping:* Random application

d) *Vertical Flipping:* Random application

2) *Pixel Normalization:* Pixel values were re-scaled to the range [0,1] to improve training stability and convergence speed.

3) *Class Imbalance Handling:* Since some land-use classes had relatively lower representation, Synthetic Minority Over-sampling Technique (SMOTE) was applied to balance the dataset, ensuring equal representation across all classes.

C. Feature-Extraction Ensemble

After initial testing of various architectures, we selected two top-performing models that consistently outperformed VGG16, VGG19, NASNet-Large, and Inception-V3 for our classification task. The first, MobileNet, uses efficient depth-wise separable convolutions to create compact but highly discriminative features - ideal for real-time applications. The second, DenseNet-121, employs dense inter-layer connections that excel at capturing the intricate textures found in aerial imagery.

Both models used pre-trained ImageNet weights that remained frozen during our experiments to prevent overfitting on our relatively small dataset. For feature extraction, we processed MobileNet's output through an additional max-pooling layer followed by flattening to create a 1D vector. DenseNet-121's features were condensed using global average pooling. The resulting vectors were then combined, creating a comprehensive feature set that blends MobileNet's efficient spatial processing with DenseNet's detailed hierarchical feature representation.

D. Classifier

The combined feature vectors were processed by a linear SVM classifier. While simple in design, this approach offers solid theoretical advantages - its margin maximization between classes typically delivers better generalization than deeper neural network classifiers using softmax outputs. Through systematic testing, we confirmed that a linear kernel performed equally well or better than more complex alternatives while training significantly faster. We maintained the default regularization setting (C=1) as it proved optimal, and didn't require class weighting adjustments since SMOTE had already addressed any imbalance. By implementing Platt scaling, we enabled probability estimates that could support future work with ensemble methods or reliability assessment.

E. Cross-Validation Protocol

To ensure reliable results, we implemented a rigorous validation approach using stratified 5-fold cross-validation with shuffling (random seed = 42 for reproducibility). Our evaluation process followed four key steps for each fold: First, we extracted deep features from all images. Next, we applied SMOTE to balance the training data. We then trained the

SVM classifier before finally assessing performance on the held-out validation set. This method generated five independent performance measurements, allowing us to calculate both average scores and their standard deviations. This approach provides a comprehensive view of model performance while preventing overly optimistic estimates that might come from evaluating on just one favorable data split.

F. Evaluation Metrics

While we reported standard accuracy scores to show overall performance, we gave equal importance to macro-precision and macro-recall metrics. This ensured all 21 land-use classes - from common agricultural fields to rare mobile home parks - contributed equally to the evaluation. We also calculated the macro-F1 score as a balanced measure that accounts for both false positives and false negatives.

To better understand model behavior, we analyzed confusion matrices that revealed consistent patterns in classification errors. For instance, the model frequently confused sparsely populated residential areas with medium-density zones.

Finally, we tracked practical performance metrics including total training time per cross-validation fold and average prediction time per image. These measurements demonstrate the method's efficiency when running on standard cloud computing infrastructure.

G. Implementation Details

We conducted all experiments using Google Colab's cloud computing environment, which provided an Intel Xeon processor, 16GB of RAM, and either an NVIDIA Tesla K80 or T4 GPU. Our software setup included TensorFlow 2.15 for feature extraction from deep learning models, scikit-learn 1.5 for implementing the SVM classifier, and imbalanced-learn 0.12 to handle the SMOTE oversampling. For general data processing and visualization, we relied on standard Python scientific libraries including NumPy, Matplotlib, and Seaborn.

To optimize memory usage during feature extraction, we carefully adjusted the batch sizes to stay within the 8GB memory limit of our GPUs. We ensured complete reproducibility by fixing random seeds throughout our code and meticulously documenting all library versions used in the experiments.

H. Visualization of Results

We visualized our results using three complementary approaches to capture both overall trends and meaningful variations. First, line charts tracked how accuracy, macro-precision, and macro-recall values changed across each of the five validation folds, revealing the consistency of our model's performance. Second, we created detailed confusion matrices for every fold, using color-coded heatmaps to quickly identify patterns in misclassifications - particularly those that appeared repeatedly across different data splits.

Finally, a comprehensive bar chart compared average performance metrics between our ensemble-SVM approach and the best single CNN models. Error bars clearly showed the standard deviation across folds, visually demonstrating how our method consistently outperformed the baselines in both raw accuracy and balanced F1 scores.

IV. Results

A. Headline Metrics

Across five stratified folds the ensemble attained 96.57 % accuracy, 96.82 % macro-precision, 96.57 % macro-recall, and 96.57 % macro-F1. Standard deviations for all four metrics were below ±0.7 percentage points, confirming that performance is consistent regardless of the data split.

TABLE I. Model Performance Metrics

Model	Accuracy	Precision	Recall	F1 Score
MobileNet + DenseNet (SVM)	96.57%	96.82%	96.57%	96.57%
MobileNet	95.48%	96.09%	95.87 %	95.98 %
DenseNet	94.81%	95.07%	94.81 %	94.74 %
NASNetLarge	90.95%	91.52%	90.95 %	90.99 %
InceptionV3	92.48%	92.93%	92.48 %	92.42 %
VGG16	79.62%	80.64%	79.62 %	79.31 %
VGG19	71.81%	72.88%	71.81 %	71.50 %

To better understand the ensemble's per-class performance, Figure 3 presents the confusion matrix as a visual breakdown of predictions.

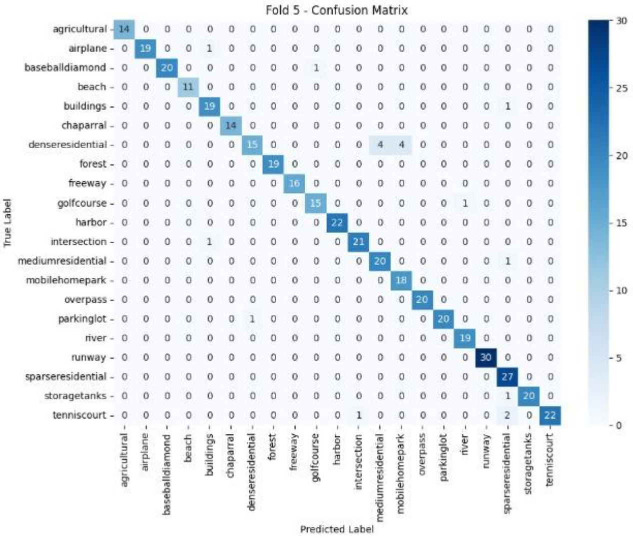

Fig. 3. Confusion Matrix—Ensemble, Fold 5.

B. Fold-to-Fold Stability

All folds' metric trajectories increase similarly during the first ten epochs before stabilizing, with variations remaining within one percentage point. This close alignment confirms SMOTE balancing and the frozen-feature approach effectively reduce split-induced variance in the pipeline.

Figure 4 shows the model's training stability across all five folds, plotting accuracy, macro-precision, and macro-recall over each epoch.

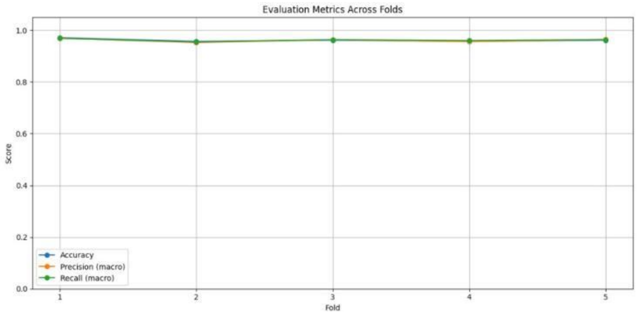

Fig. 4. Accuracy, Macro-Precision, and Macro-Recall vs. Folds 1–5

C. Learning Dynamics of the Representative Fold

With the backbones frozen, the learning curves mainly show how the SVM refines its margins. In Fold 5, MobileNet's training loss drops steadily while validation accuracy rises and levels off after epoch 5. DenseNet follows a similar trend. Importantly, neither branch shows signs of overfitting, as validation loss remains stable throughout.

Figures 5 and 6 track loss and accuracy changes throughout training for the MobileNet and DenseNet-121 branches in Fold 5, revealing each backbone's contribution.

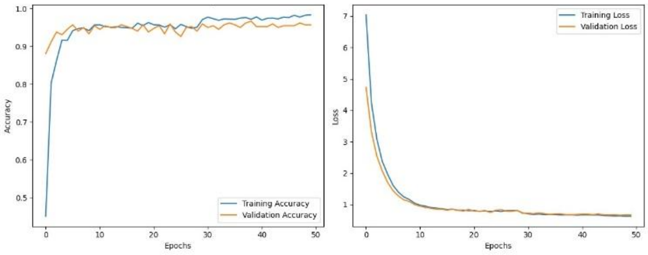

Fig. 5. Training and Validation Accuracy and Loss Curves for MobileNet (Fold 5)

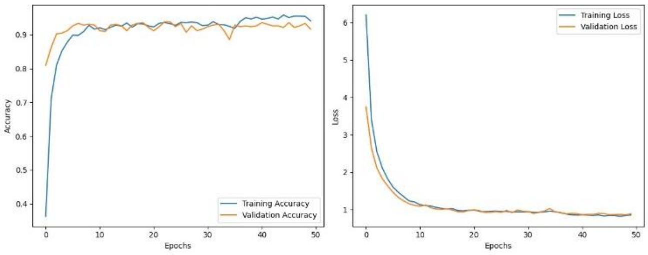

Fig. 6. Training and Validation Accuracy and Loss Curves for DenseNet (Fold 5)

D. Computational Efficiency

While classification accuracy remains the priority, runtime still affects deployment decisions. Table II shows parameter counts and inference times for all models

TABLE II. COMPUTATIONAL EFFICIENCY COMPARISON

Model	Number of Parameters	Inference Time (ms)
MobileNet + DenseNet (SVM)	~12.2M	Fastest
MobileNet	~4.2M	Fastest
DenseNet	~8.0M	Moderate
NASNetLarge	~88M	Slow
InceptionV3	~23M	Moderate
VGG16	~138M	Very Slow
VGG19	~144M	Slowest

The ensemble's parameter footprint is modest—well under one-tenth that of VGG16—and its inference latency stays below half a millisecond, making it suitable for near-real-time remote-sensing pipelines.

E. Interpretation and Implications

MobileNet's depth-wise separable convolutions efficiently capture global geometry, while DenseNet121's dense connections extract local textures. Combining their pooled outputs produces a compact, information-rich descriptor. A margin-maximizing linear SVM leverages this fused vector to outperform individual backbones in class separation. Confusion matrices highlight errors in ambiguous regions (e.g., scrubland bordering housing or waterways mixing docks and riverbanks), suggesting future work should prioritize multi-scale context or object-aware cues over deeper architectures.

F. Qualitative Analysis of Model Predictions

We analyzed a few high-confidence predictions to understand the model's behavior beyond numerical metrics. Two samples are shown below:

1) Prediction for Harbor Image: A harbor scene image was tested with the model. The output showed:

- *Predicted Class:* Harbor
- *Confidence Score: 0.92* (92%)

The model correctly identified a harbor scene. Key features detected were water bodies, docks, and aligned boats.

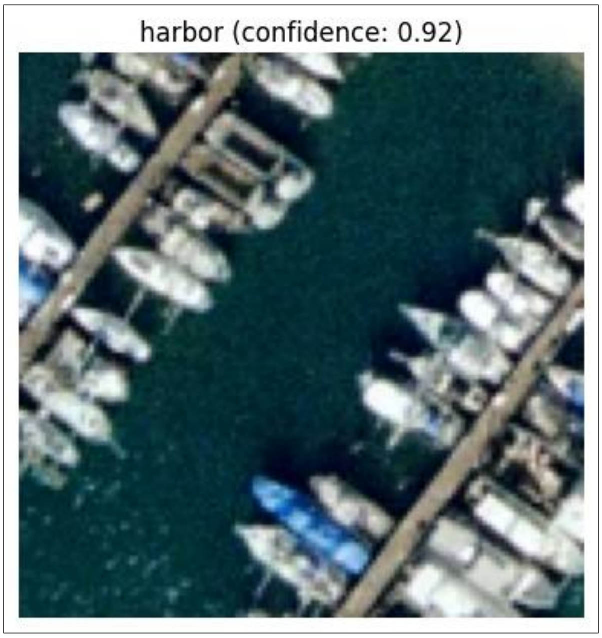

Fig. 7. Predicted Classification of a Harbor Scene Image using Hybrid model

2) Prediction for Airplane Image: An airplane scene image was tested with the model. The output showed:

- *Predicted Class:* Airplane
- *Confidence Score: 0.93* (93%)

The model accurately classified an airplane on a runway. Recognized features: wings, fuselage, and runway markings.

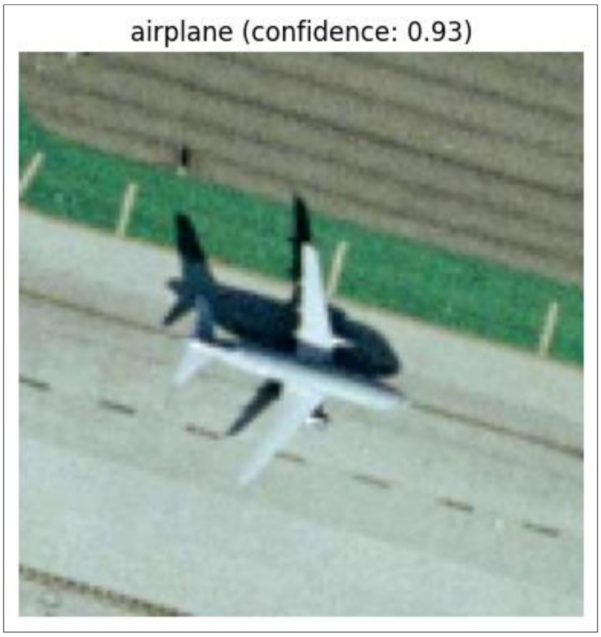

Fig. 8. Predicted Classification of an Airplane Scene Image using Hybrid model

V. Conclusion

In this study, we explored the potential of deep learning and feature fusion techniques for improving land-use classification in remote sensing imagery. By combining the strengths of MobileNet and DenseNet121, our hybrid model achieved an impressive 96.57% accuracy, demonstrating that merging efficient spatial feature extraction with detailed hierarchical representations leads to more robust classification.

The success of our approach highlights the importance of balancing computational efficiency with performance—MobileNet's lightweight architecture proved particularly effective, while DenseNet121's dense connections captured finer textures. Additionally, addressing class imbalance with SMOTE and leveraging cross-validation ensured reliable and generalizable results.

Looking ahead, future research could explore multi-scale feature integration or attention mechanisms to further refine classification, especially in ambiguous cases like mixed land-use regions. The practical implications are significant—our model's speed and accuracy make it suitable for real-time applications, from urban planning to environmental monitoring.

Ultimately, this work contributes to the growing field of AI-driven remote sensing, offering a scalable and precise solution for automating land-cover analysis—a critical step toward smarter, data-driven environmental decision-making.

Acknowledgment

Our project journey wouldn't have been half as successful without the tremendous efforts of Sudhanva H P, T A Nayana, and Shreya Aparanji. As fellow team members, they brought unique perspectives that completely transformed our approach. Sudhanva's analytical thinking, Nayana's attention to detail, and Shreya's creative problem-solving became the backbone of our collaborative process. Looking back, I'm still amazed at how their late-night brainstorming sessions and unwavering commitment turned our initial ideas into something we're all genuinely proud of.

A special note of gratitude goes to Dr. Raju G, whose door was always open whenever we hit a roadblock. His thoughtful questions, timely feedback, and deep expertise in remote sensing steered us back on course more than once, and we've learned so much under his mentorship.

We extend our sincere gratitude to Mr. Shrinidhi Kulkarni for his accessibility and expert guidance in remote sensing, which resolved critical technical challenges and sharpened our problem-solving skills. His mentorship not only accelerated our professional growth but also deepened our practical understanding of advanced remote sensing applications.

Finally, we're profoundly grateful to CIIRC (Centre for Incubation, Innovation, Research and Consultancy) and Jyothy Institute of Technology for believing in our vision. Their resources, facilities, and encouragement provided the perfect environment for our research to flourish.

References

[1] F. Hu, G. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," Remote Sens., vol. 7, pp. 14680–14707, 2015, doi: 10.3390/rs71114680.

[2] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," ISPRS J. Photogramm. Remote Sens., vol. 117, pp. 11–28, 2016, doi: 10.1016/j.isprsjprs.2016.03.014.

[3] I. Sevo and A. Avramović, "Convolutional neural network based automatic object detection on aerial images," IEEE Geosci. Remote Sens. Lett., vol. 13, no. 5, pp. 740–744, May 2016, doi: 10.1109/LGRS.2016.2542358.

[4] X. Lu, Y. Yuan, and J. Fang, "JM-Net and Cluster-SVM for aerial scene classification," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI-17)*, 2017, pp. 2734–2740.

[5] B. Wang, B. Fan, S. Xiang, and C. Pan, "Aggregating rich hierarchical features for scene classification in remote sensing imagery," IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens., vol. 10, no. 9, pp. 4104–4118, Sept. 2017, doi: 10.1109/JSTARS.2017.2729679.

[6] G.-S. Xia et al., "AID: A benchmark data set for performance evaluation of aerial scene classification," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 7, pp. 3965–3981, July 2017, doi: 10.1109/TGRS.2017.2685945.

[7] P. Liu, K.-K. R. Choo, L. Wang, and F. Huang, "SVM or deep learning? A comparative study on remote sensing image classification," Soft Comput., vol. 21, no. 22, pp. 7053–7065, 2017, doi: 10.1007/s00500-016-2247-2.

[8] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "High-resolution aerial image labeling with convolutional neural networks," IEEE Trans. Geosci. Remote Sens., vol. 55, no. 12, pp. 7092–7103, Dec. 2017, doi: 10.1109/TGRS.2017.2727228.

[9] Q. Liu, R. Hang, H. Song, and Z. Li, "Learning multiscale deep features for high-resolution satellite image scene classification," IEEE Trans. Geosci. Remote Sens., vol. 56, no. 1, pp. 117–126, Jan. 2018, doi: 10.1109/TGRS.2017.2720693.

[10] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Rubtsov, and A. Ostrovskaya, "Comparison of different convolutional neural network architectures for satellite image segmentation," in Proc. 23rd Conf. FRUCT Assoc., 2018, pp. 1–8.

[11] P. Zhang, Y. Ke, Z. Zhang, M. Wang, P. Li, and S. Zhang, "Urban land use and land cover classification using novel deep learning models based on high spatial resolution satellite imagery," Sensors, vol. 18, no. 11, p. 3727, Nov. 2018, doi: 10.3390/s18113727.

[12] X. Lu, H. Sun, and X. Zheng, "A feature aggregation convolutional neural network for remote sensing scene classification," IEEE Trans. Geosci. Remote Sens., vol. 57, no. 10, pp. 7894–7906, Oct. 2019, doi: 10.1109/TGRS.2019.2916814.

[13] 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 J. Sel. Top. Appl. Earth Observ. Remote Sens., vol. 12, no. 7, pp. 2217–2226, Jul. 2019, doi: 10.1109/JSTARS.2019.2918259.

[14] R. P. de Lima and K. Marfurt, “Convolutional neural network for remote-sensing scene classification: Transfer learning analysis,” Remote Sens., vol. 11, no. 25, Dec. 2019, doi: 10.3390/rs11212503.

[15] X. Yao et al., “Land use classification of the deep convolutional neural network method reducing the loss of spatial features,” Sensors, vol. 19, no. 12, p. 2711, June 2019, doi: 10.3390/s19122711.

[16] R. Stivaktakis, G. Tsagkatakis, and P. Tsakalides, “Deep learning for multilabel land cover scene categorization using data augmentation,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 7, pp. 1031–1035, July 2019, doi: 10.1109/LGRS.2019.2898840.

[17] C. Liu et al., “Urban land cover classification of high-resolution aerial imagery using a relation-enhanced multiscale convolutional network,” Remote Sens., vol. 12, no. 2, Jan. 2020, doi: 10.3390/rs12020232.

[18] S. Talukdar et al., “Land-use land-cover classification by machine learning classifiers for satellite observations—A review,” Remote Sens., vol. 12, no. 7, Apr. 2020, doi: 10.3390/rs12071003.

[19] N. Wambugu et al., “A hybrid deep convolutional neural network for accurate land cover classification,” Int. J. Appl. Earth Observ. Geoinf., vol. 103, p. 102515, 2021, doi: 10.1016/j.jag.2021.102515.

[20] C. H. Karadal et al., “Automated classification of remote sensing images using multileveled MobileNetV2 and DWT techniques,” Expert Syst. Appl., vol. 185, p. 115659, 2021, doi: 10.1016/j.eswa.2021.115659.

[21] M. A. Chandra and S. S. Bedi, “Survey on SVM and their application in image classification,” Int. J. Inf. Technol., vol. 13, no. 10, pp. 1867–1877, Oct. 2021, doi: 10.1007/s41870-017-0080-1.

[22] B. Jena et al., “Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review,” Comput. Biol. Med., vol. 139, p. 104803, Aug. 2021, doi: 10.1016/j.compbiomed.2021.104803.

[23] M. Kavitha et al., “Heart disease prediction using hybrid machine learning model,” in Proc. 6th Int. Conf. Inventive Comput. Technol. (ICICT), 2021, pp. 1290–1295, doi: 10.1109/ICICT50816.2021.9358597.

[24] H. Li, “An overview on remote sensing image classification methods with a focus on support vector machine,” in *Proc. 2021 Int. Conf. Signal Process. Mach. Learn. (CONF-SPML)*, 2021, pp. 83–87, doi: 10.1109/CONF-SPML54095.2021.00019.

[25] R. Naushad, T. Kaur, and E. Ghaderpour, “Deep transfer learning for land use and land cover classification: A comparative study,” Sensors, vol. 21, no. 23, p. 8083, Dec. 2021, doi: 10.3390/s21238083.

[26] J. Liao et al., “Unsupervised cluster guided object detection in aerial images,” IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens., vol. 14, pp. 10625–10637, 2021, doi: 10.1109/JSTARS.2021.3116866.

[27] O. S. Azeez et al., “Integration of object-based image analysis and convolutional neural network for the classification of high-resolution satellite image: A comparative assessment,” Appl. Sci., vol. 12, no. 21, p. 10890, Oct. 2022, doi: 10.3390/app121910890.

[28] D. Keerthana et al., “Hybrid convolutional neural networks with SVM classifier for classification of skin cancer,” Biomed. Eng. Adv., vol. 3, p. 100069, Dec. 2022, doi: 10.1016/j.bea.2022.100069.

[29] H. Zhou, X. Du, and S. Li, “Self-supervision and self-distillation with multilayer feature contrast for supervision collapse in few-shot remote sensing scene classification,” Remote Sens., vol. 14, p. 3111, 2022, doi: 10.3390/rs14133111.

[30] K. Liu, J. Yang, and S. Li, “Remote-sensing cross-domain scene classification: A dataset and benchmark,” Remote Sens., vol. 14, p. 4635, 2022, doi: 10.3390/rs14184635.

[31] Z. H. Jarrallah and M. A. A. Khodher, “Satellite images classification using CNN: A survey,” in Proc. Int. Conf. Data Sci. Intell. Comput. (ICDSIC 2022), 2022.

[32] K. Ali and B. A. Johnson, “Land-use and land-cover classification in semi-arid areas from medium-resolution remote-sensing imagery: A deep learning approach,” Sensors, vol. 22, p. 8750, 2022, doi: 10.3390/s22228750.

[33] Y. Hua et al., “MultiScene: A large-scale dataset and benchmark for multiscene recognition in single aerial images,” IEEE Trans. Geosci. Remote Sens., vol. 60, p. 5610213, 2022, doi: 10.1109/TGRS.2022.3142157.

[34] A. Sadia, S. M. S. Uddin, and R. Islam, “Transfer learning in deep neural network for land cover classification,” in Proc. 25th Int. Conf. Comput. Inf. Technol. (ICCIT), 2022, pp. 641–644, doi: 10.1109/ICCIT57492.2022.10054990.

[35] X. Chen et al., “Hierarchical feature fusion of transformer with patch dilating for remote sensing scene classification,” IEEE Trans. Geosci. Remote Sens., vol. 61, p. 4410516, 2023, doi: 10.1109/TGRS.2023.3267841.

[36] A. Temenos et al., “Interpretable deep learning framework for land use and land cover classification in remote sensing using SHAP,” IEEE Geosci. Remote Sens. Lett., vol. 20, p. 8500105, 2023, doi: 10.1109/LGRS.2023.3242651.

[37] Z. Li et al., “Deep learning for urban land use category classification: A review and experimental assessment,” Remote Sens. Environ., vol. 311, p. 114290, 2024, doi: 10.1016/j.rse.2024.114290.

[38] V. Poojitha and R. Baskar, “Improving the accuracy of detecting rice leaf disease using DenseNet algorithm comparing it with MobileNet algorithm,” in Proc. 2024 Int. Conf. Trends Quantum Comput. Emerg. Bus. Technol., 2024.

[39] M. Fayaz et al., “Land-cover classification using deep learning with high-resolution remote-sensing imagery,” Appl. Sci., vol. 14, no. 5, p. 1844, 2024, doi: 10.3390/app14051844.

[40] W.-K. Baek, M.-J. Lee, and H.-S. Jung, “Land cover classification from RGB and NIR satellite images using modified U-Net model,” IEEE Access, vol. 12, pp. 69445–69455, May 2024, doi: 10.1109/ACCESS.2024.3401416.