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

FusionNet-RS: A Deep Feature Fusion Model for Remote Sensing Image Classification on PatternNet

Shreya Aparanji
Department of *Artificial Intelligence and Machine Learning*
Jyothy Institute of Technology
Bengaluru, India
shreya.aparanji12@gmail.com

T. A. Nayana
Department of *Computer Science and Engineering*
Jyothy Institute of Technology
Bengaluru, India
nayana.ta2003@gmail.com

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

Sudhanva H. P
Department of *Computer Science and Engineering*
Jyothy Institute of Technology
Bengaluru, India
hpsudhanva2@gmail.com

Dr. Raju G
Ex- ISRO
Project Director
Megha Tropicques
Bengaluru, India
raju_garuda@yahoo.co.in

Dr. Madhu B. R
Department of *Artificial Intelligence and Machine Learning*
Jyothy Institute of Technology
Bengaluru, India
madhu.br@jyothyit.ac.in

***Abstract*—Remote sensing image classification plays a crucial role in various applications, including land cover mapping and urban planning. Traditional classification methods struggle with complex spatial patterns, necessitating advanced deep learning techniques for improved accuracy and efficiency. Convolutional Neural Networks (CNNs) have demonstrated remarkable success in remote sensing, offering automated feature extraction and superior classification performance. However, selecting an optimal CNN architecture remains challenging due to trade-offs between computational efficiency and accuracy. This study proposes a deep learning model integrating MobileNet and DenseNet121 to enhance classification performance on the PatternNet dataset. MobileNet's lightweight structure ensures computational efficiency, while DenseNet121's deep feature extraction improves classification accuracy. A five-fold cross-validation approach was employed to evaluate the model's robustness. Experimental results indicate that the feature fusion model achieves a mean accuracy of 95.98%, outperforming individual architectures in both accuracy and generalization.**

Keywords—Remote Sensing, Image Classification, Convolutional Neural Networks (CNNs), PatternNet Dataset, Feature Fusion Model

I. INTRODUCTION

Convolutional Neural Networks (CNNs) automatically extract spatial and spectral information from satellite or aerial images to classify remote sensing data accurately[1]. This approach enhances classification accuracy by helping CNNs recognize complex patterns in the data [4]. Models like MobileNet and DenseNet shows outstanding traits with respect to improvement in the performance on large datasets [4]. Transfer learning is one method that uses pre-trained models to achieve this [2]. Data augumentation methods like rotation and scaling helps in the improvement of model's ability to generalize[8]. Performance evaluation uses metrics like accuracy, precision, recall, and F1-score to ensure the robustness of models [13]. CNNs simplify the classification of remote sensing data and supporting tasks such as environmental monitoring, urban planning, and land cover mapping [5].

The PatternNet dataset is a well-organised collection of aerial images designed for supervised image classification in remote sensing [7]. The dataset contains high-resolution aerial images divided into four land-use categories: agriculture, industry, residential areas and woodland [7]. PatternNet contains labeled images, making it ideal for training and testing deep learning models for land-use classification and image analysis [8]. The dataset's diverse and organised categories make it a popular choice for testing deep learning models like VGG, ResNet, DenseNet and MobileNet in aerial image classification [9]. The dataset provides a suitable environment for applying advanced machine learning techniques such as transfer learning, data enhancement, and feature extraction. This is due to its inherent characteristics of high variability within classes and similarity between classes, which pose considerable challenges [12]. Researchers using the Patternet dataset generate land-cover maps, strengthen environmental monitoring, and refine city planning [5].Cited in [3], the study uses the PatternNet dataset, a benchmark made up of high-resolution aerial photographs of several land cover categories.Model generalisation capacity was improved by means of data standardisation, data enhancement, and SMOTE class imbalance management [18].To guarantee robustness and avoid overfitting, a further five-fold cross-validation technique was used [8].

Using feature fusion to combine different deep learning models can make classification more accurate. It also helps in creating faster and better image classification models. MobileNet and DenseNet are widely used for remote sensing image classification because they extract features effectively. MobileNet speeds up processing with its small design and DenseNet spreads features more effectively. Feature fusion combines the strengths of both models to build a classification model with higher accuracy. Global Average Pooling (GAP) extracts important features from each model and combines them into one final output. Then, L2 regularization, dropout and fully connected layers are used to refine this final result. Tests showed MobileNet achieved 87.34% accuracy and DenseNet achieved 90.21%. When combined, their accuracy increased to 93.87%. This big improvement shows how combining features can boost classification accuracy in remote sensing tasks like land-use mapping, environmental monitoring, urban planning, disaster management, and crop monitoring.

II. LITERATURE SURVEY

Deep learning methods like CNNs have greatly improved remote sensing image classification. Methods like Random Forests and SVMs often missed complex patterns since they depend on manually selected features instead of learning from the images themselves. CNNs perform better by automatically learning patterns at multiple levels which leads to more accurate results [2], [8]. Studies show that deep CNNs outperform traditional methods as they can effectively extract spatial and contextual details from remote sensing images [1], [7].

Pre-trained CNN models are widely used in remote sensing because they extract features effectively. Many applications have used models like VGGNet, ResNet, DenseNet, MobileNet, and NASNet [2], [8]. AlexNet which was released in 2012, was the first deep learning model to improve image classification. ResNet added residual connections in 2016, making it easier to train deep networks and avoid vanishing gradient issues [7]. DenseNet (2017) improves feature reuse by letting information flow through its dense connections [26]. MobileNet is known for its small size, making it ideal for devices with limited processing power [4]. While individual CNN models are well studied in remote sensing, researchers are still exploring how combining different architectures can improve results [12].

To fully benefit from the advantages of various architectures, hybrid CNN models have been developed.By investigating techniques that combine both sophisticated and straightforward machine learning models, research has attempted to improve classification accuracy while preserving computational efficiency.It was shown that combining MobileNet and ResNet improved accuracy and decreased computational complexity [7].A recent study also tested a hybrid VGG-DenseNet model [24], demonstrating that DenseNet's feature reuse capabilities improved classification accuracy.Prior research has demonstrated that feature fusion techniques, which entail joining feature vectors, can improve classification results by integrating complementary information from several designs [21]. Despite advancements, it is still difficult to fine-tune these feature fusion models because large-scale remote sensing projects require a balance between processing speed and accuracy [12].

The following sections will cover the specifics of the model selection, training strategy, data preparation techniques, and evaluation metrics. This entails a thorough analysis of the outcomes and a performance comparison.

III. METHODOLOGY

A structured technique was used to assess deep learning models' classification performance on the PatternNet dataset in a methodical manner. This covers model selection, managing class imbalance, data preparation, and a sound training plan. Each stage of the experimental pipeline is described in depth in the next subsections.

A. Dataset Description

The PatternNet dataset, a reference dataset for classifying remote sensing images, is used in this work. The collection includes 38 classes with pictures of different structural complexity, each of which represents a unique land-use pattern. To guarantee uniformity in input dimensions among models, the photos were scaled to 224 by 224 pixels.

Figure 1 shows images from all the different classes in the PatternNet dataset.

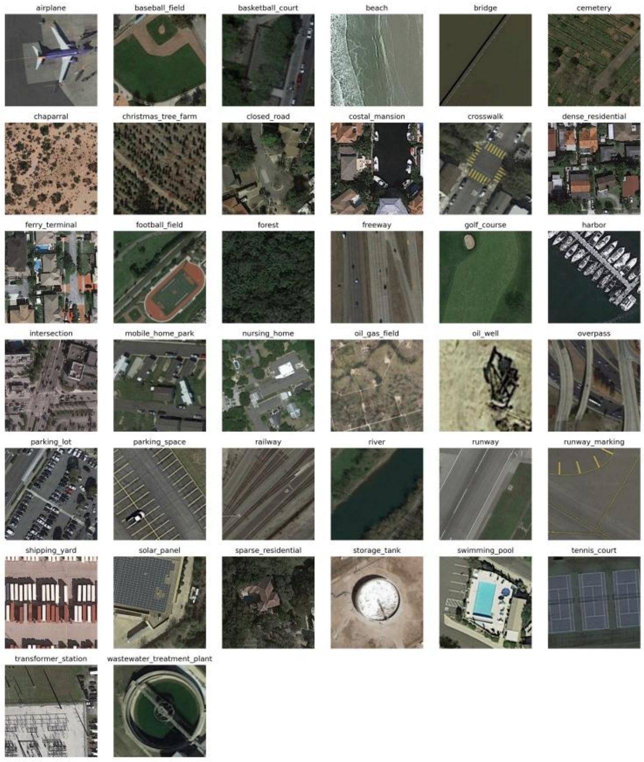

Fig. 1. Sample images from all classes in the PatternNet dataset

B. Data Preprocessing

A number of preprocessing procedures were carried out to improve the quality of the input photos and speed up the model's learning process in order to get the dataset ready for training.

1) Data Normalization: To scale all of the photos between [0,1], they were converted to float32 and normalized by dividing the pixel values by 255.0.

2) One-Hot Encoding: One-hot encoding was used to change the categorical class labels, producing a multi-class vector representation.

3) Data Augmentation: To enhance model generalization, the training dataset was augmented using:

- Random rotations (10 degrees)
- Width and height shifts (10%)
- Horizontal flips

C. Handling Class Imbalance

The training data was subjected to the Synthetic Minority Over-sampling Technique (SMOTE) in order to rectify the class imbalance. Before returning to the original 3D format, the input pictures were molded into a 2D feature space and synthetic examples were created for underrepresented classes.

D. Model Selection and Architecture

Two pre-trained Convolutional Neural Networks (CNNs), namely MobileNet and DenseNet121, were used as feature extractors due to their efficiency in remote sensing applications.

1) Data Feature Extraction:

- Both models were initialized with ImageNet weights.

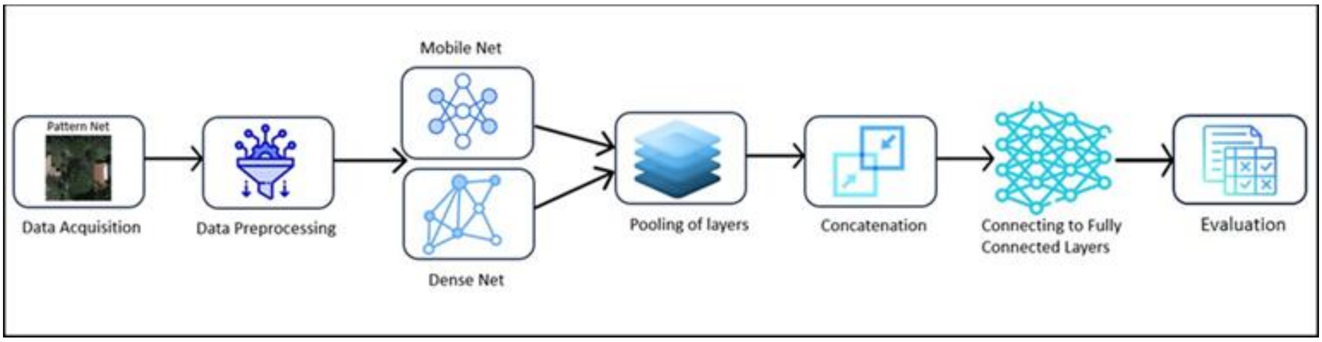

Fig. 2. Architecture flowchart of MobileNet - DenseNet model

- The features were extracted from the base model, and a Global Average Pooling (GAP) layer was applied to obtain compact feature representations.
- The models' parameters were frozen to leverage pre-trained knowledge and avoid overfitting.

2) Feature Fusion:

- The extracted feature vectors from both models were concatenated to form a unified feature representation.

3) Classification Head:

- Random A fully connected dense layer (256 units, ReLU activation) with L2 regularization (λ=0.01).
- A dropout layer (40%) to mitigate overfitting
- A final dense layer (128 units, ReLU activation, L2 regularization) followed by another dropout layer (30%).
- A softmax output layer with 38 neurons (corresponding to the number of classes)

E. Training and Evaluation Strategy

A systematic training and assessment process was used to guarantee the model's efficacy and generalizability. The methods used to maximize model performance and avoid overfitting are described in this section.

1) Cross-Validation: A 5-fold cross-validation (KFold) strategy was employed to ensure robustness and generalization across different subsets of data.

2) Loss Function and Optimizer: The Categorical Cross-Entropy loss function, which works well for multi-class classification issues, was used to create the model. The Adam optimizer, which ensures adaptive learning rate modifications to improve convergence and stability, was used to optimize the training process using a learning rate of 0.0005.

3) Learning Rate Scheduling: ReduceLROnPlateau was implemented to reduce the learning rate by a factor of 0.5 when validation loss plateaued for 5 epochs.

4) Early Stopping and Model Checkpointing: To prevent overfitting and ensure optimal model performance, early stopping was implemented, halting training when the validation loss failed to improve for seven consecutive epochs. Additionally, model check pointing was employed to save the best-performing model weights based on validation accuracy, ensuring that the most effective model was retained for evaluation.

F. Performance Metrics

By analyzing the models' performance across a number of criteria, a thorough evaluation of their classification accuracy and generalizability was carried out.Accuracy was the primary metric used to evaluate forecast accuracy across different classifications.Loss curve analysis was made possible by plotting training and validation loss patterns, which provided insight into potential overfitting and model convergence.In order to assess the model's generalization effectiveness, trends in validation accuracy were monitored over a wide range of folds.

G. Computational Evaluation

To guarantee efficient training and assessment, every experiment was carried out in a cloud-based Google Colab environment that made use of GPU acceleration. TensorFlow 2.x was used to implement the models, making use of the available neural network optimization tools.

In order to speed up complex calculations, the system's technical specifications included an NVIDIA Tesla K80 or T4 graphics processing unit, an Intel Xeon CPU, and 16GB of RAM. The model was constructed and visualized using libraries like Matplotlib, scikit-learn, NumPy, and TensorFlow 2.x.

MobileNet demonstrated a faster training time per fold when compared to DenseNet121, demonstrating its effectiveness and reduced computational cost.DenseNet121, on the other hand, is the recommended choice for applications needing precision because it achieved higher accuracy despite the longer training period.MobileNet was lightweight and showed lower memory usage, which made it ideal for scenarios with limited resources, in contrast to DenseNet121, which used more resources but offered a more favorable efficiency-performance balance.

In terms of scalability and implementation feasibility, the models varied.MobileNet is better suited for real-time applications and edge devices because of its quicker training and lower resource consumption, while DenseNet121, which uses more power, is better suited for high-precision tasks where accuracy is crucial.

IV. Results

A cross-validation method was used to assess the hybrid CNN model's classification performance. Key measures including as accuracy, precision, recall, and F1-score were used to evaluate the fusion model, which combined the MobileNet and DenseNet121 extractors.

A. *Overall Performance Metrics*

The results of the cross-validation show how reliable and successful the feature fusion model is in classifying the PatternNet dataset. The model performed consistently well across several folds, with a mean accuracy of 0.9598 and a standard deviation of ±0.0031.

Likewise, there was little variation in the mean accuracy, recall, and F1-score, which were 0.9472, 0.9561, and 0.9493, respectively. These outcomes demonstrate how well the feature fusion model learns discriminative characteristics and generates precise predictions.

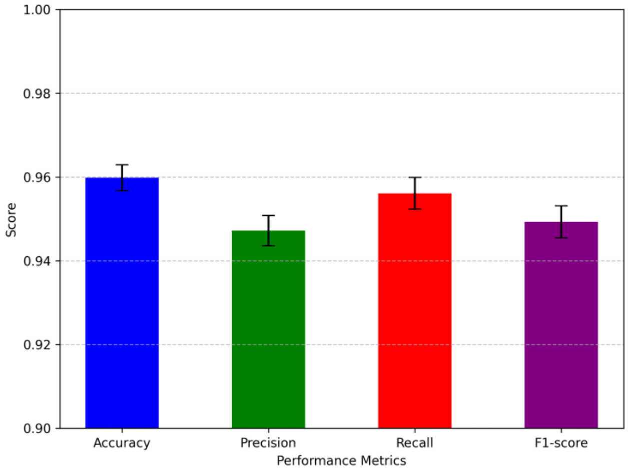

Fig. 3. Overall Performance Comparison of the feature fusion model

B. *Cross-Validation Analysis*

The cross-validation results show how well the feature fusion model classifies the PatternNet dataset and how resilient it is. With a mean accuracy of 0.9598 and a standard deviation of ±0.0031, the model demonstrated consistently great performance across various folds.

Likewise, the F1-score, mean accuracy, and mean recall were all found to have little variation, at 0.9473, 0.9561, and 0.9493, respectively. These outcomes demonstrate the feature fusion model's effectiveness in identifying discriminative characteristics and producing precise forecasts.

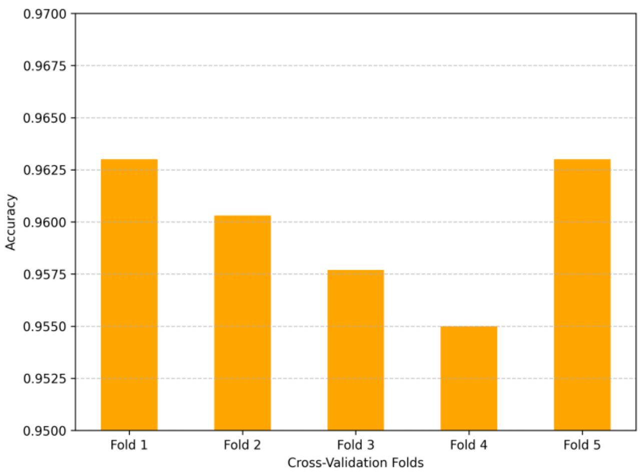

Fig. 4. Fold-wise Accuracy Performance of the feature fusion model

C. *Comparative Analysis*

The feature fusion model achieved higher accuracy and more reliable classification by combining both networks, instead of using MobileNet or DenseNet121 alone. The model gave better results by combining MobileNet's speed and simple design with DenseNet's strong feature extraction.

The model showed better performance and high efficiency. It used fewer resources and still achieved high accuracy which made it both effective and cost friendly. This balance of accuracy and efficiency makes the fusion approach a practical choice for large-scale remote sensing applications.

D. *Convergence and Training Stability*

The model learned effectively during training, with validation accuracy becoming stable after a few epochs. A steady drop in the loss showed that the learning process was well-optimized.

To avoid overfitting and keep the model general, regularization methods like dropout and L2 regularization were used. Early stopping was also applied to stop training once validation accuracy stopped improving, helping avoid extra training.

By avoiding abrupt fluctuations in the model's performance, the adaptive learning rate decay also helped to ensure steady convergence. The model adjusted its parameters and improved classification accuracy by decreasing the learning rate when progress slowed.

E. *Performance Across Different Categories*

The feature fusion model ensured balanced classification results by maintaining good recall and precision across the majority of categories. The model's capacity to identify well-represented patterns was strengthened by the near-perfect accuracy attained by categories with a large number of training data, such as vegetation and metropolitan areas.

The hybrid model improved accuracy for minority classes like bare land and small water bodies, which are often misclassified due to uneven data. SMOTE was used during training to create realistic minority class samples and help the model recognize underrepresented classes better.

Similar-looking classes like buildings and roads had some misclassification due to shared texture and color. Techniques like contrastive learning could help better classify these classes with more fine-tuning.

Future research could explore attention mechanisms to help the model focus on important visual areas and improve accuracy in challenging cases. Multi-scale feature extraction could also help capture small differences between similar-looking classes.

F. *Performance of Individual Models*

5-fold cross-validation on the PatternNet dataset was used to evaluate MobileNet and DenseNet121's classification performance separately. MobileNet obtained precision, recall, and F1-score values of 94.40%, 95.37%, and 94.58%, respectively, and an average accuracy of 95.37%. Its lightweight design made training more effective and reduced computing costs, although it showed minor differences in accuracy between folds.

The average accuracy of DenseNet121, on the other hand, was 92.33%, with precision of 92.07%, recall of 92.33%, and

F1-score of 91.67%. Its strong connectivity helped capture richer features but sometimes increased the risk of overfitting.. Both MobileNet's better generalization and DenseNet121's robust feature extraction are highlighted in these assessments; the feature fusion model used both to improve classification results.

G. Visualization of Model Performance

Monitoring accuracy and loss patterns provide important information about the model's stability and learning process during training. Analyzing cross-validation folds ensures dependable performance in remote sensing image categorization by evaluating the model's generalization ability and spotting any problems like overfitting or underfitting. These patterns are depicted in Figures 5–9, which provide a clear visual depiction of the behavior and efficacy of the model.

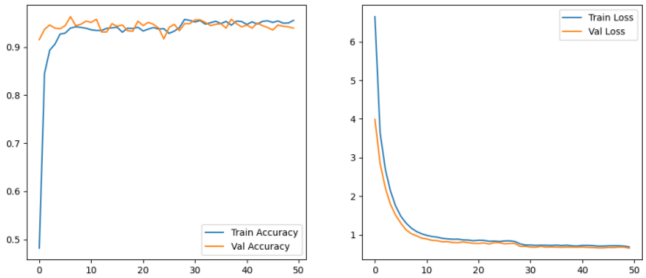

Fig. 5. Training and Validation Accuracy and Loss Curves for Fold 1

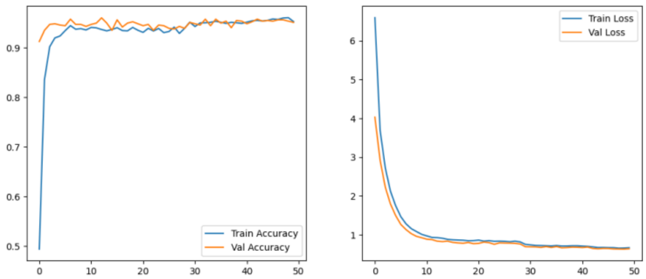

Fig. 6. Training and Validation Accuracy and Loss Curves for Fold 2

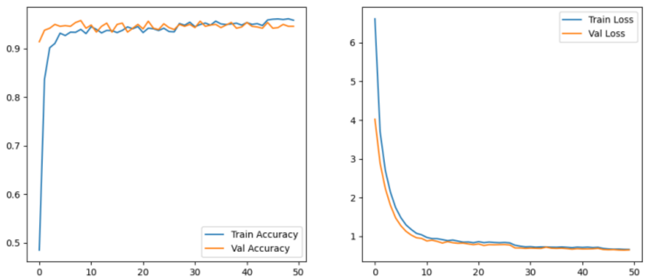

Fig. 7. Training and Validation Accuracy and Loss Curves for Fold 3

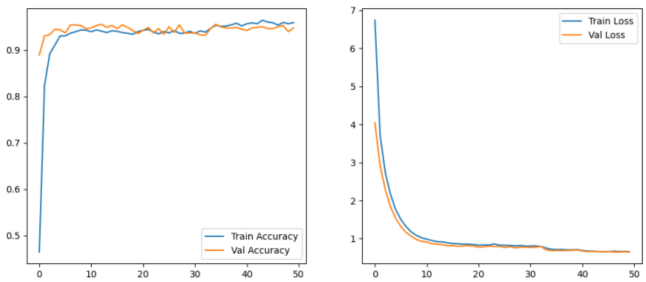

Fig. 8. Training and Validation Accuracy and Loss Curves for Fold 4

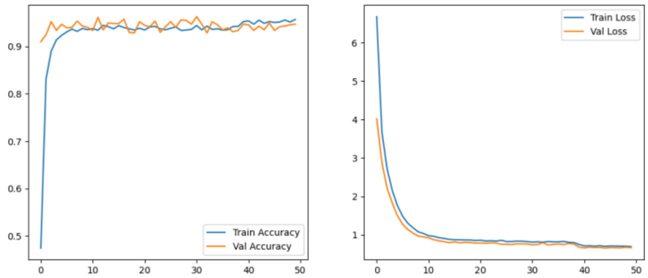

Fig. 9. Training and Validation Accuracy and Loss Curves for Fold 5

H. Qualitative Analysis of Model Predictions

The predict_image function was used to pick and analyze test pictures from the PatternNet dataset in order to evaluate the trained model's performance in the real world. When input photos are processed, predictions are made, and the classification results and confidence ratings are displayed. The model's accuracy in classifying unseen pictures is demonstrated by the test cases that follow.

1) Prediction for Baseball Field Image: A test image of a baseball field was fed into the trained model. The image was preprocessed by resizing it to the required dimensions and normalizing the pixel values. After passing through the model, the output showed:

- *Predicted Class:* Baseball Field
- *Confidence Score:* 1.00 (100%)

The model's ability to correctly identify the distinguishing characteristics of a baseball field, including its diamond shape, green turf, and dirt base paths, was validated by the visualization. The model appears to have successfully learnt the patterns related to this category, based on the high confidence score.

Fig. 10. Predicted Classification of a Baseball Field Image Using CNN Model

2) Prediction for Christmas Tree Farm Image: In another test, an image of a Christmas tree farm was used to evaluate the model’s classification accuracy. Similar preprocessing steps were applied, and the model produced the following results:

- *Predicted Class:* Christmas Tree Farm
- *Confidence Score:* 0.97 (97%)

The model accurately recognized the main characteristics of a Christmas tree farm, such as a natural setting and rows of trees that are uniformly spaced. A solid forecast with high dependability is shown by the confidence score, even if it was somewhat lower than in the baseball field scenario.

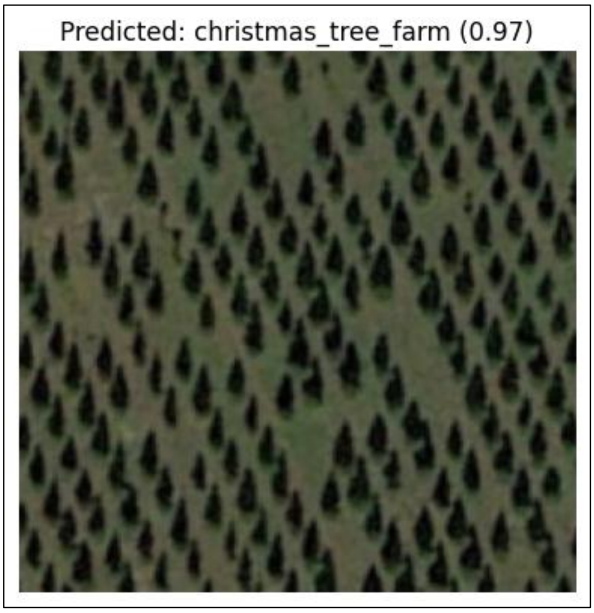

Fig. 11. Predicted Classification of a Christmas Tree Farm Image Using CNN Model

In summary, the high confidence values of 1.00 and 0.97 indicate that the model performed well in classification on the PatternNet dataset. Accurate predictions were produced thanks in large part to DenseNet121's sophisticated feature extraction capabilities, which preserved spatial features. The little decline in confidence raises the possibility that some photos share characteristics with other categories or have a higher visual complexity, which calls for more research to identify possible misclassifications.

I. Model Training Time and Computational Efficiency

Evaluating the feature fusion model's performance in large-scale applications requires knowing how long it takes to train and how much processing power it needs. Because it can balance processing speed and accuracy, the MobileNet-DenseNet121 design is a good choice for classifying remote sensing images. Training time, GPU use, and memory usage all play a big role in how practical the model is for real-world use.

The feature fusion model balanced speed and performance by combining DenseNet121’s strong feature extraction with MobileNet’s lightweight design. MobileNet made computation faster, while DenseNet121 improved accuracy, though with slightly longer training times. This balance made the model a good choice for tasks where accuracy matters more than speed.

MobileNet used less memory while DenseNet121 used deeper layers for better results. Overall memory and GPU usage stayed within limits. The model handled resources well during training and kept costs low. Learning rate scheduling and batch normalization helped it train efficiently. Techniques like model trimming and quantization could make it even more efficient without reducing accuracy.

V. Conclusion

Using the PatternNet dataset, the primary goal of this study was to evaluate the effectiveness of a hybrid CNN model that combines MobileNet and DenseNet121 in classification tasks. The dataset offers detailed aerial images of diverse land cover types and is widely used in remote sensing research. The feature fusion model was designed to extract features better and classify images accurately.

The experimental results showed that the feature fusion model performed well, achieving an average accuracy of 95.98% using five-fold cross-validation. Combining MobileNet and DenseNet121 provided a good balance between complex features and efficiency. The model trained smoothly and handled class imbalance effectively with SMOTE. Overall, it outperformed individual models in classifying aerial images.

In addition feature selection areas of study might focus on improving the model by adding attention mechanisms. Additionally, examining transformer-based models and self-supervised learning strategies may improve classification accuracy even more. Furthermore, evaluating the model's performance on real-world remote sensing applications and expanding the dataset to include a wider variety of land cover categories will support future research.

Acknowledgment

For their essential contributions to our project, we would like to express our sincere gratitude to our other team members, Shrihari V Pandurangi, Tejas M Bharadwaj, Yashas N, and Varshini L. We are appreciative of their efforts, since their commitment, diligence, and teamwork have substantially enhanced the project.

We also want to sincerely thank Jyothy Institute of Technology and CIIRC (Centre For Incubation, Innovation, study and Consultancy) for allowing us to conduct our study on remote sensing and for providing the tools and assistance we needed.

We would like to express our sincere gratitude to Shrihari V Pandurangi, Tejas M Bharadwaj, Yashas N, and Varshini L, who have made significant contributions to our project.Their dedication, hard work, and collaborative effort have significantly improved the project, and we are grateful for their contributions.

We would like to extend our sincerest gratitude to Jyothy Institute of Technology and CIIRC (Centre For Incubation, Innovation, Research and Consultancy) for permitting us to carry out our research on remote sensing and for furnishing us with the necessary equipment and support.

References

[1] F. Hu et al., "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," Remote Sens., vol. 7, pp. 14680-14707, Nov. 2015, doi: 10.3390/rs71114680.

[2] X. X. Zhu et al., "A comprehensive review and list of resources: Deep learning in remote sensing," IEEE Geosci. Remote Sens. Mag., vol. 5, Dec. 2017.

[3] J. L. Wang et al., "Classification of white blood cells with PatternNet-fused ensemble of convolutional neural networks (PECNN)," in Proc. IEEE Conf., 2018.

[4] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," in Proc. IEEE Conf., Indianapolis, USA, 2019.

[5] P. Helber et al., "EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 12, no. 7, pp. 2217–2226, Jul. 2019.

[6] G. Chen et al., "GT-DeepLabv3+: A deep learning-based algorithm for field crop segmentation," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 12, no. 7, Jul. 2019.

[7] M. YoungK. Xu et al., "Multilayer Feature Fusion Network for Scene Classification in Remote Sensing," IEEE Geosci. Remote Sens. Lett., vol. 17, no. 11, Nov. 2020.

[8] S. Gupta et al., "Remote Sensing Image Classification using Deep Learning," in Proc. 10th Int. Conf. Syst. Model. Adv. Res. Trends (SMART), Moradabad, India, Dec. 2021.

[9] Z. H. Jarrallah and M. A. A. Khodher, "Satellite Images Classification Using CNN: A Survey," in Proc. Int. Conf. Data Sci. Intell. Comput. (ICDSIC), Karbala, Iraq, 2022, doi: 10.1109/ICDSIC56987.2022.10075828.

[10] L. Chen et al., "EFCNet: Ensemble full convolutional network for semantic segmentation of high-resolution remote sensing images," IEEE Geosci. Remote Sens. Lett., vol. 19, p. 8011705, 2022.

[11] Y. Liao et al., "EFLDet: Enhanced Feature Learning for Object Detection," Neural Comput. Appl., vol. 34, pp. 1033–1045, 2022, doi: 10.1007/s00521-021-06607-1.

[12] G. Rousset et al., "Assessment of Deep Learning Techniques for Land Use Land Cover Classification in Southern New Caledonia," Remote Sens., vol. 13, no. 12, p. 2257, Jun. 2021, doi: 10.3390/rs13122257.

[13] Q. Li et al., "Performance Evaluation of Deep Learning Classification Network for Image Features," IEEE Access, vol. 9, Jan. 2021, doi: 10.1109/ACCESS.2020.3048956.

[14] Y. Zhou and C.-Y. Su, "Efficient convolutional neural network for pest recognition - ExquisiteNet," in Proc. 2nd IEEE Eurasia Conf. IoT, Commun. Eng. (ECICE), 2020.

[15] J. Xu, X. Wang, W. Wang, and W. Huang, "PHCU-Net: A parallel hierarchical cascade U-Net for skin lesion segmentation," Biomed. Signal Process. Control, [Online]. Available: www.elsevier.com/locate/bspc.

[16] L. Jiang et al., "A lightweight spatially-aware classification model for breast cancer pathology images," Biocybern. Biomed. Eng., [Online]. Available: www.elsevier.com/locate/bbe.

[17] S. Pan et al., "Land-cover classification of multispectral LiDAR data using CNN with optimized hyper-parameters," ISPRS J. Photogramm. Remote Sens., [Online]. Available: www.elsevier.com/locate/isprsjprs.

[18] A. Ozdemir et al., "Classification of imbalanced hyperspectral images using SMOTE-based deep learning methods," Expert Syst. Appl., [Online]. Available: www.elsevier.com/locate/eswa.

[19] H. Li and R. Ramachandra, "Deep learning-based fingerprint presentation attack detection: A comprehensive survey," Institute of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU), Norway.

[20] H. Liu et al., "Google is all you need: Semi-supervised transfer learning strategy for light multimodal multi-task classification model," arXiv preprint arXiv:2501.01611, Jan. 2025.

[21] J. Nie et al., "Cross-modal feature fusion and interaction strategy for CNN-Transformer-based object detection in visual and infrared remote sensing imagery," IEEE Geosci. Remote Sens. Lett., vol. 21, p. 5000405, 2024.

[22] H. Wu et al., "CTFNet: CNN-Transformer fusion network for remote-sensing image semantic segmentation," IEEE Geosci. Remote Sens. Lett., vol. 21, p. 5000305, 2024.

[23] L. Kong et al., "DP-ProtoNet: An interpretable dual path prototype network for medical image diagnosis," in Proc. IEEE 22nd Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom), 2023, doi: 10.1109/TrustCom60117.2023.00390.

[24] 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.

[25] Y. Huang et al., "Remote Sensing Object Counting Through Regression Ensembles and Learning to Rank," IEEE Trans. Geosci. Remote Sens., vol. 61, p. 4701617, 2023.

[26] 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. (TQCEBT), Pune, India, Mar. 2024.

[27] Y. Alkhurayyif and A. R. Wahab Sait, "Deep learning-driven dyslexia detection model using multi-modality data," PeerJ Comput. Sci., vol. 10, no. e2077, Jun. 2024, doi: 10.7717/peerj-cs.2077.

[28] F. Liu et al., "Vision-based environmental perception for autonomous driving," Proc. IMechE, Part D: J. Automob. Eng., vol. 239, no. 1, pp. 39–69, 2025.

[29] 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, Jul. 2017, doi: 10.1109/TGRS.2017.2685945.

[30] W. Zhou, "Dataset," Accessed: Mar. 19, 2025. [Online]. Available: https://sites.google.com/view/zhouwx/dataset.