International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ENHANCED LAND USE AND LAND COVER CLASSIFICATION USING WEIGHTED ENSEMBLE LEARNING OF PRETRAINED CNN MODELS

Jayesh Dhanesha[1], Dr. Sweta Panchal[2]

[1]Research Scholar, Department of Computer Engineering, Dr Subhash University, Gujarat, India

[2]Associate Professor, Department of Electronics & Communication Engineering, Dr Subhash University, Gujarat, India.

Abstract

Accurate Land Use and Land Cover (LULC) classification is essential for environmental monitoring, precision agriculture, urban planning, and disaster management. Although Convolutional Neural Networks (CNNs) have significantly advanced remote sensing image interpretation, their individual performance varies across heterogeneous land-cover categories due to architectural differences and representational biases. To address these inconsistencies, this study proposes an enhanced ensemble learning framework that integrates six pretrained CNN architectures—VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV2—fine-tuned on the UC Merced Land Use dataset. Three ensemble strategies were examined: simple averaging, soft voting, and weighted averaging based on grid-search optimization. Extensive experiments were conducted to evaluate classification accuracy, F1-score, inference time, and GPU memory consumption. DenseNet121 achieved the highest individual accuracy (97.05%), while the weighted averaging ensemble significantly outperformed all standalone models, reaching 99.43% accuracy and an F1-score of 0.993 with efficient inference performance. These findings demonstrate that combining diverse CNN architectures enhances classification robustness, reduces model-specific weaknesses, and improves stability across visually similar LULC categories. The proposed ensemble framework offers a scalable, high-accuracy solution suitable for real-world Earth observation tasks and contributes a comprehensive comparative evaluation of ensemble strategies for LULC classification using deep transfer learning.

Keywords: Land Use and Land Cover (LULC); Convolutional Neural Networks (CNNs); Deep Learning; Ensemble Learning; Transfer Learning

1. Introduction

Understanding Land Use and Land Cover (LULC) patterns is fundamental to a wide range of geospatial applications, including environmental monitoring, precision agriculture, urban planning, and disaster management. High-resolution remote sensing imagery provides an effective means of observing land dynamics at scale. However, its efficient interpretation requires robust computational models capable of handling complex spatial variations. Traditional machine learning approaches—such as Support Vector Machines, Random Forests, and k-Nearest Neighbors—depend heavily on handcrafted spectral and textural features, which

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

limit their ability to perform accurately in heterogeneous landscapes characterized by irregular textures, varying illumination, and noise (Belgiu & Drăguț, 2016; Shao et al., 2019).

The emergence of deep learning, and particularly Convolutional Neural Networks (CNNs), has transformed remote sensing image analysis by enabling hierarchical feature extraction directly from raw imagery (Zhong et al., 2016). Pretrained CNN architectures such as VGG16, ResNet50, DenseNet121, InceptionV3, EfficientNetB0, and MobileNetV2 have demonstrated substantial performance gains through transfer learning, leveraging learned ImageNet features to improve downstream classification tasks (He et al., 2016; Huang et al., 2017; Sandler et al., 2018; Tan & Le, 2019). Despite their success, individual architectures often exhibit variable performance across different LULC categories due to differences in network depth, connectivity patterns, receptive fields, and representational biases. For example, a model optimized for object-level features may perform well in urban categories but struggle with visually similar natural classes such as forests and chaparral.

Ensemble learning provides a principled solution to such inconsistencies by integrating multiple models to leverage their complementary strengths, reduce variance, and mitigate architecture-specific weaknesses (Dietterich, 2000; Ghosh et al., 2020). While previous studies have explored CNN ensembles for LULC classification, many employ only a limited set of models or prioritize accuracy without addressing computational considerations such as inference time, memory utilization, and scalability (Dou et al., 2024; Khatami et al., 2022). Moreover, systematic comparisons across diverse ensemble fusion strategies—such as soft voting, simple averaging, and performance-based weighted averaging—remain limited.

To address these gaps, this study investigates a deep ensemble learning framework integrating six pretrained CNN architectures and evaluates three fusion strategies to enhance LULC classification on the UC Merced dataset. The objective is to establish a robust, scalable, and high-accuracy classification system while providing a detailed comparison of computational efficiency metrics. By leveraging the complementary representational capabilities of diverse CNN architectures, the proposed ensemble approach aims to improve classification stability, reduce architectural bias, and advance the practical applicability of deep learning for real-world remote sensing workflows.

2. Related Work

Land Use and Land Cover (LULC) classification has advanced substantially with the progression of remote sensing technologies and deep learning methodologies. Traditional machine learning classifiers such as Support Vector Machines, Random Forests, and k-Nearest Neighbors rely heavily on handcrafted spectral or textural features, which constrain their ability to generalize across heterogeneous spatial environments (Belgiu & Drăguț, 2016; Shao et al., 2019). These approaches often fail when confronted with intra-class variability, complex spatial textures, and variations in illumination, motivating the shift toward automated feature extraction techniques.

A. Deep Learning for LULC Classification

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Deep learning, particularly Convolutional Neural Networks (CNNs), has transformed remote sensing image analysis by enabling pixel-level feature learning and hierarchical representation extraction (Ma et al., 2019; Zhu et al., 2017). CNNs have proven effective in representing both local spatial details and global semantic features, consistently outperforming classical hand-engineered approaches (Zhong et al., 2016).

Transfer learning further enhances performance when annotated remote sensing data are limited. Pretrained models initialized with ImageNet weights—including VGG16 (Simonyan & Zisserman, 2015), ResNet50 (He et al., 2016), DenseNet121 (Huang et al., 2017), InceptionV3 (Szegedy et al., 2016), EfficientNetB0 (Tan & Le, 2019), and MobileNetV2 (Sandler et al., 2018)—enable rapid convergence and improved generalization across diverse land-cover categories (Helber et al., 2019). These architectures differ significantly in depth, connectivity, receptive fields, and parameter efficiency, resulting in unique representational biases that can lead to inconsistent performance across different LULC classes.

B. Ensemble Learning in Remote Sensing

Ensemble learning techniques integrate predictions from multiple models to enhance robustness and reduce variance (Dietterich, 2000). In remote sensing, ensemble strategies such as bagging, boosting, voting, and weighted averaging have demonstrated consistent improvements in land-cover classification, particularly in datasets characterized by high inter-class similarity or complex appearance variability (Ghosh et al., 2020; Dou et al., 2024).

Recent studies have applied ensembles to tasks such as vegetation mapping, multi-spectral image classification, and high-resolution LULC categorization (Zhong et al., 2016; Khatami et al., 2022). However, many previous works combine only a small number of CNN models or focus primarily on accuracy without addressing operational metrics including computational cost, inference time, or GPU memory usage—factors that significantly affect deployment feasibility in real-world applications.

C. Research Gaps

A review of existing literature highlights three major limitations that this study addresses:

1. **Limited Diversity of Architectures:** Previous ensemble studies commonly integrate only two or three CNNs (e.g., ResNet + VGG or Inception + ResNet), which restricts complementary feature extraction potential (Zhang et al., 2019).
2. **Lack of Systematic Comparison Across Fusion Methods:** Few studies evaluate multiple ensemble strategies—such as soft voting, simple averaging, and performance-based weighted fusion—within a unified experimental framework (Dou et al., 2024).
3. **Insufficient Reporting of Computational Efficiency:** Many existing works emphasize accuracy alone, neglecting essential operational metrics such as inference latency, GPU memory consumption, and training complexity (Khatami et al., 2022).

By employing six pretrained CNN architectures and evaluating three ensemble fusion strategies with comprehensive classification and operational metrics, this research provides a more holistic assessment of ensemble deep learning for LULC classification.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. Methodology

3.1 Dataset Description

All experiments in this study were conducted using the UC Merced Land Use Dataset, a widely used benchmark for high-resolution aerial image classification. The dataset includes 2,100 RGB images across 21 balanced land-use categories, each with a spatial resolution of 256 × 256 pixels at approximately 0.3m (Yang & Newsam, 2010). The classes span diverse landscapes—including agricultural fields, airports, residential areas of various densities, industrial structures, transportation networks, water bodies, and recreational facilities—

captured

Figure 1. Sample images for each of the 21 land-use classes from the UC Merced dataset

across several metropolitan regions within the United States. Figure 1 presents sample images representing each class. This geographic and visual diversity makes the dataset highly suitable for evaluating model generalization to heterogeneous land-cover types (Zhao et al., 2023). For robust model development, the dataset was partitioned following widely adopted conventions into 70% for training, 20% for validation, and 10% for testing, ensuring an unbiased and representative evaluation protocol.

3.2 Preprocessing Pipeline

Proper preprocessing is essential for aligning remote sensing imagery with the input distribution expected by ImageNet-pretrained CNN models. Each architecture in this study required normalization consistent with its original training configuration. Models such as VGG16 and ResNet50 were preprocessed using BGR channel ordering combined with per-channel mean subtraction derived from ImageNet statistics (Simonyan & Zisserman, 2015; He et al., 2016). In contrast, InceptionV3, EfficientNetB0, DenseNet121, and MobileNetV2

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

required pixel-value scaling to the range [−1,1] following their respective standard ImageNet preprocessing pipelines (Szegedy et al., 2016; Tan & Le, 2019; Huang et al., 2017).

To enhance generalization and mitigate overfitting, data augmentation was applied during training to improve generalization. The augmentation strategy included random geometric transformations such as horizontal and vertical flips, rotations up to ±15°, spatial shifts, and controlled zoom variations. Additionally, random cropping followed by resizing to the original input dimensions introduced further spatial variability. These augmentation operations, widely regarded as effective in deep learning for remote sensing, simulate realistic variations in scale, orientation, and illumination, thereby improving model robustness (Shorten & Khoshgoftaar, 2019; Aldahoul et al., 2023).

3.3 Transfer Learning Strategy

Transfer learning has become a standard approach in remote sensing due to the limited availability of large annotated datasets (Ma et al., 2019). In this study, six pretrained CNN architectures—VGG16, ResNet50, InceptionV3, DenseNet121, EfficientNetB0, and MobileNetV2—were initialized with ImageNet weights (Deng et al., 2009). Figure 2 illustrates the transfer learning architecture employed in this work, adapted from Simonyan and Zisserman (2015), He et al. (2016), and Tan and Le (2019). The original classification heads were removed and replaced with a unified architecture tailored for the 21-class LULC task.The redesigned head comprised a Global Average Pooling layer for spatial feature aggregation, followed by a dense layer with 512 ReLU units, a dropout layer with a rate of 0.2 to reduce overfitting, and a final softmax layer to produce class probabilities.

All models were fine-tuned using an identical training protocol to ensure comparability. The Adam optimizer was employed due to its adaptive learning rate capabilities, with categorical cross-entropy as the loss function. Training was conducted with a batch size of 32 for a maximum of 50 epochs. To prevent overfitting and ensure stable convergence, ReduceLROnPlateau was used to dynamically reduce the learning rate when validation performance plateaued, while early stopping with a patience value of five epochs halted training in the absence of further validation accuracy gains.

Based on validation accuracy and architectural diversity, four models—ResNet50, EfficientNetB0, DenseNet121, and MobileNetV2—were selected for ensemble construction. These architectures were chosen because they provided strong predictive performance while capturing complementary representational characteristics. The remaining models (VGG16 and InceptionV3) were retained as comparative baselines.

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

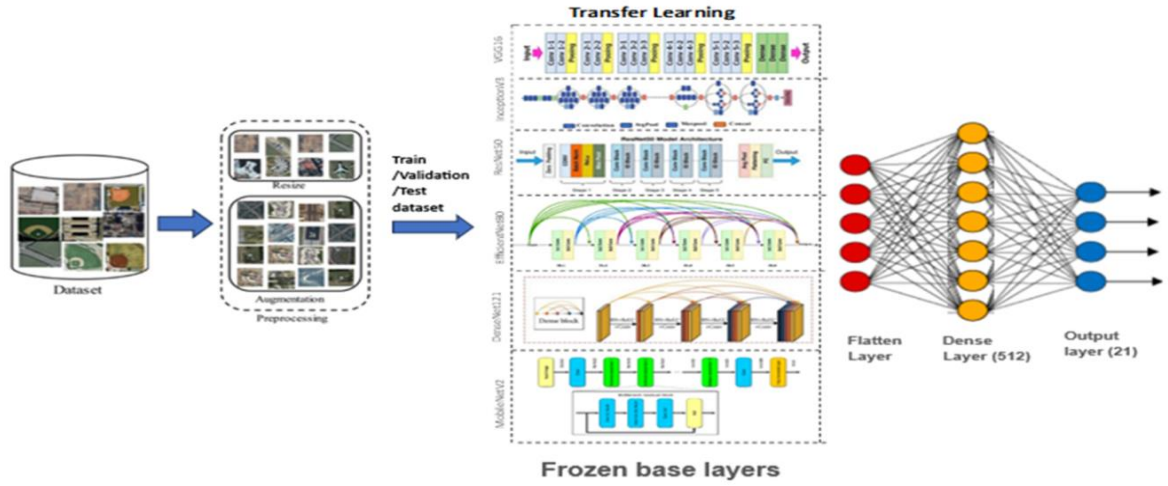

Figure 2: Transfer learning strategy

3.4 Ensemble Framework

Ensemble learning integrates predictions from multiple base classifiers to improve robustness, reduce variance, and enhance generalization performance (Dietterich, 2000). In this study, three ensemble fusion strategies were implemented to combine the outputs of the four selected CNN architectures: simple averaging, soft voting, and weighted averaging.

3.4.1 Simple Averaging

In simple averaging, the ensemble prediction is obtained by computing the arithmetic mean of the probability distributions generated by the individual models. For a test sample x, the averaged probability for class c is defined as:

$$P_{avg}(c) = \frac{1}{N}\sum_{i=1}^{N} \quad P_i(c) \ldots\ldots\ldots \text{Equation (1)}$$

where N=4 is the number of base CNN models and $P_i(c)$ is the predicted probability of class c by model i. This method assumes equal contribution from all models and reduces prediction variance.

3.4.2 Soft Voting (Probability-Based Decision)

Soft voting selects the final class label based on the highest averaged probability across all models:

$$\hat{C} = argmax\,(P_i(c)) \ldots\ldots\ldots Equation\ (2)$$

Soft voting is particularly effective when base classifiers provide well-calibrated probability estimates and share complementary error patterns.

3.4.3 Weighted Averaging (Performance-Based Fusion)

Weighted averaging assigns different importance levels to each model based on their validation performance. The weighted ensemble probability for class c is computed as:

$$P_{weighted}(c) = \sum_{i=1}^{N} \quad w_i P_i(c) \;\; subject\ to: \sum_{i=1}^{N} \quad w_i = 1 \ldots\ldots\ldots \text{Equation (3)}$$

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A grid search with a step size of 0.1 was performed on the validation set to identify the optimal weight configuration. The resulting optimal weights were: ResNet50 (0.4), EfficientNetB0 (0.4), DenseNet121 (0.1), and MobileNetV2 (0.1). This performance-based weighting emphasizes models with stronger predictive accuracy while still incorporating architectural diversity.

All ensemble strategies were implemented in Python using NumPy for efficient vectorized computation. Their effectiveness was evaluated in terms of classification accuracy and computational metrics, including inference time and GPU memory utilization, to assess real-world deployment feasibility.

3.5 Evaluation Metrics

A combination of classification and computational metrics was used to provide a comprehensive evaluation of model performance. Classification performance was assessed using accuracy, precision, recall, and F1-score, computed in both macro-averaged and weighted-averaged forms. These metrics are widely used for multiclass remote sensing problems because they capture model behavior across balanced and imbalanced classes (Khatami et al., 2022; Ma et al., 2019).

To assess operational feasibility, inference time per image was measured by processing all test samples on an NVIDIA GPU and computing the mean processing duration per image. GPU memory usage was monitored using the NVIDIA System Management Interface to quantify peak memory consumption during inference. These computational metrics are critical for assessing real-world deployability, particularly in environments with limited hardware resources.

4. Experimental Setup and Configuration

This section describes the computational environment, training procedures, and ensemble configuration used to evaluate the proposed LULC classification framework. The experimental setup was carefully designed to ensure reproducibility and to provide a comprehensive performance comparison across individual CNN architectures and ensemble strategies.

4.1 Computational Environment

All experiments were conducted on the Kaggle cloud computing platform using an NVIDIA Tesla P100-PCIE-16GB GPU. This hardware configuration offers sufficient computational capacity for training deep convolutional neural networks, enabling efficient batch processing and parallel model execution. The platform ensured reliable runtime performance and consistent environmental conditions across all experiments.

4.2 Model Training and Convergence Analysis

Following the transfer learning strategy outlined in Section 3.3, all six CNN architectures were trained using identical hyperparameters to maintain a fair comparison. The training configuration is summarized in Table 1.

Table 1: Training specifications for individual CNN models

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Training Parameters	**Value**
Maximum Epochs:	50
Optimizer:	Adam
Loss Function	Categorical cross-entropy
Batch Size	32
Early Stopping	patience = 5 epochs (monitoring validation accuracy)

Each model was trained independently using the 1,470-image training subset, with performance continuously monitored on the 420-image validation set. Training was terminated early if validation accuracy plateaued according to the early stopping criterion. The best-performing model checkpoint (lowest validation loss) was retained for final evaluation and ensemble construction. The validation performance and training characteristics of each architecture are presented in Table 2.

Table 2: Validation performance and training characteristics of individual CNN architectures

Model	**Epochs**	**Train Accuracy**	**Val Accuracy**	**Training Time (sec)**	**Train Memory Used (MB)**
VGG16	50	0.9166	0.9462	5151.75	2384.37
InceptionV3	42	0.9441	0.9643	7742.04	3795.9
ResNet50	22	0.941	0.9738	2442.01	2801.91
EfficientNetB0	24	0.9686	0.9743	2527.94	2575.63
DenseNet121	28	0.9657	0.9724	3109.81	2721.93
MobileNetV2	26	0.9524	0.9681	2769.48	2346.38

The convergence analysis reveals several notable trends. EfficientNetB0 achieved the highest validation accuracy (97.43%) with rapid convergence in only 24 epochs. DenseNet121 achieved the lowest validation loss (0.0703), indicating strong confidence calibration despite slightly lower accuracy. ResNet50 demonstrated the fastest convergence, completing training in 22 epochs with relatively low computational cost. In contrast, InceptionV3 required longer training (42 epochs, 7,742 seconds), likely due to its multi-scale architectural design. VGG16

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

underperformed across all metrics, aligning with expectations given its older and less expressive architecture compared to modern CNN designs.

4.3 Model Selection for Ensemble Construction

Based on validation accuracy and architectural diversity, four models, ResNet50, EfficientNetB0, DenseNet121, and MobileNetV2—were selected to construct the ensemble. These models complement one another through varied architectural strengths: ResNet50 contributes deep residual learning; EfficientNetB0 provides optimized compound scaling; DenseNet121 enhances feature reuse through dense connectivity; and MobileNetV2 offers lightweight, efficient feature extraction via depthwise separable convolutions. The remaining two models, VGG16 and InceptionV3, served as baseline comparisons but were excluded from the ensemble due to their comparatively lower performance or higher computational cost.

4.4 Ensemble Fusion Strategies

Three ensemble fusion strategies, as described in Section 3.4, were systematically implemented using the four selected models. Figure 3 illustrates the weighted averaging ensemble architecture.

To identify the most effective combination of models for the ensemble, a grid search was carried out on the validation set. The search tested multiple weight combinations in steps of 0.1, ensuring that the total weight always equalled one. For each combination, validation accuracy was evaluated. The best results were obtained when ResNet50 and EfficientNetB0 were each assigned a weight of 0.4, while DenseNet121 and MobileNetV2 were assigned weights of 0.1 each.

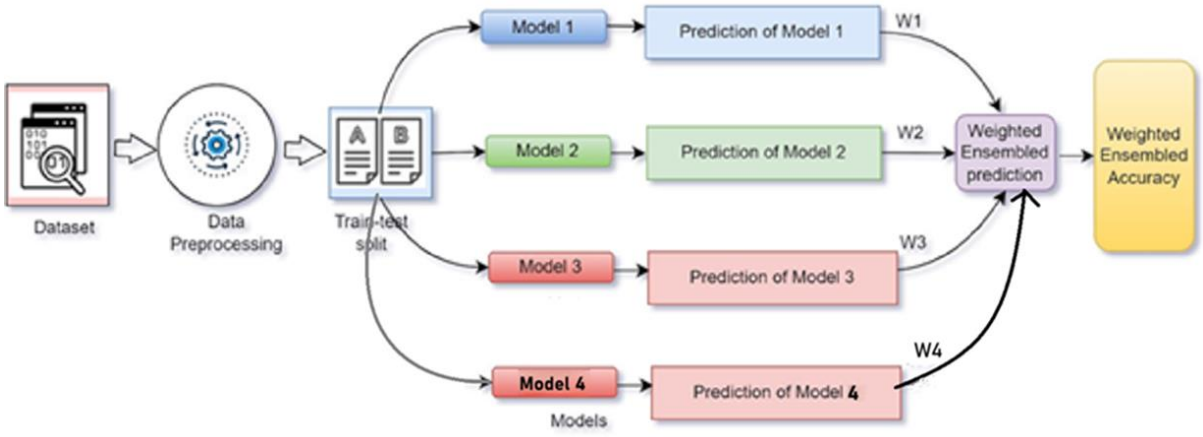

Figure 3. Proposed weighted ensemble learning architecture

This distribution indicates that the ensemble method effectively emphasized the strongest-performing models (ResNet50 and EfficientNetB0), while still benefiting from the complementary features of DenseNet121 and MobileNetV2.

4.4 Evaluation Protocol

All ensemble strategies and the six individual models—including the two baselines (VGG16 and InceptionV3)—were evaluated using the 210-image independent test set, which remained

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

entirely unseen during training and validation. This strict data separation ensured unbiased performance assessment. Evaluation included both classification quality metrics (accuracy, precision, recall, F1-score) and operational performance metrics (inference time and GPU memory consumption).

4.5 Implementation Details

All ensemble fusion strategies were implemented using custom Python scripts, with NumPy employed for efficient vectorized numerical operations. Model predictions (logits and probability vectors) were extracted from the saved checkpoints corresponding to the best validation performance. For reproducibility, random seeds were fixed across all frameworks, including Python, NumPy, and TensorFlow. The complete experimental pipeline—from data preprocessing through model training, ensemble fusion, and final evaluation—was executed sequentially to eliminate inconsistencies arising from environmental variations.

5. Results and Discussion

This section presents the performance of the individual CNN architectures and compares the results achieved by the three ensemble learning strategies on the independent test dataset. The discussion emphasizes predictive accuracy, computational efficiency, robustness, and real-world deployment considerations.

5.1 Baseline Performance of Individual CNN Architectures

The transfer learning approach provided a strong baseline across all individual models, demonstrating the effectiveness of ImageNet-pretrained weights for LULC classification. Table 3 summarizes the test accuracy, inference time, and memory consumption for each architecture

Table 3. Test set performance of individual CNN architectures

Model	Test Accuracy	Inference Time	Inference Memory used (MB)
VGG16	0.922857	32.43	1675.95
InceptionV3	0.954286	40.27	1842.83
ResNet50	0.96	29.31	1679.68
EfficientNetB0	0.96	33.02	1701.37
DenseNet121	0.970476	45.16	1836.33
MobileNetV2	0.949524	26.87	1660.81

DenseNet121 achieved the highest test accuracy (97.05%). Its dense connectivity pattern enables efficient feature reuse and stable gradient propagation, which improves its ability to

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

distinguish visually similar land-cover categories. ResNet50 and EfficientNetB0 both reached 96.00% accuracy, albeit through different architectural philosophies: residual connections in ResNet50 preserve gradient flow, while EfficientNetB0's compound scaling balances depth, width, and resolution (He et al., 2016; Tan & Le, 2019).

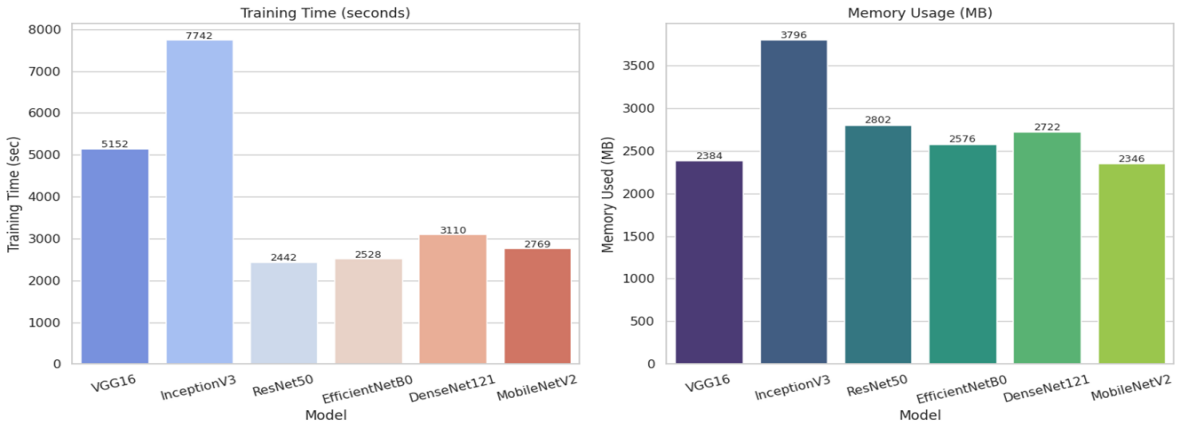

Figure 4. Training time and memory usage comparison for six CNN-based models

MobileNetV2 showed the fastest inference (26.87 seconds) and lowest memory usage, making it suitable for resource-constrained settings, although at a slight accuracy loss (94.95%). VGG16, the oldest architecture evaluated, produced the lowest accuracy (92.29%), reinforcing the benefits of modern connectivity mechanisms (Huang et al., 2017). InceptionV3 achieved strong accuracy (95.43%) but incurred higher computational cost, likely due to its multi-scale feature extraction modules. DenseNet121's high accuracy required substantially greater inference time (45.16 seconds), approximately 68% slower than MobileNetV2, highlighting the accuracy–efficiency trade-off common in deep CNNs.

5.2 Ensemble Performance Analysis

The proposed ensemble framework produced substantial gains over the best-performing individual model (DenseNet121, 97.05% accuracy). The test set results for the three fusion strategies are summarised in Table 4, which reports accuracy, inference time, and GPU memory usage for each ensemble configuration.

Table 4. Test set performance of ensemble strategies

Ensemble Approach	Test Accuracy	Inference Time (s)	Memory Usage (MB)
Averaging Ensemble	0.989524	106.695	3810.594
Weighted Averaging (GridSearch) Ensemble	0.994286	98.85775	3793.766
Soft Voting Ensemble	0.986667	98.43912	3804.832

Received: August 08, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Across all fusion methods, the ensembles consistently surpassed DenseNet121, confirming that aggregating diverse CNN architectures improves robustness and predictive power for LULC classification (Dietterich, 2000; Ghosh et al., 2020). The Soft Voting ensemble achieved 98.67% accuracy, corresponding to an absolute gain of 1.62 percentage points over DenseNet121. Simple Averaging further improved performance to 98.95%, highlighting the benefit of averaging full probability distributions instead of making class decisions solely via voting.

The Weighted Averaging ensemble with grid-search–optimized weights (0.4 for ResNet50, 0.4 for EfficientNetB0, 0.1 for DenseNet121, and 0.1 for MobileNetV2) achieved the best overall performance, reaching 99.43% test accuracy. This corresponds to an absolute improvement of 2.38 percentage points over DenseNet121 and confirms the effectiveness of performance-based weighting that emphasizes the strongest base learners while still leveraging complementary representations from the remaining models. In error-rate terms, the ensemble reduced the misclassification rate from 2.95% (DenseNet121) to 0.57%, representing an approximately 80% reduction in error.

From a computational perspective, all ensembles required higher resources than single models, as expected when four CNNs are executed in parallel. Inference time increased to roughly 98–107 seconds for the full 210-image test set, and peak GPU memory usage more than doubled to around 3,800 MB. Among the ensemble methods, Soft Voting achieved the lowest inference time (≈98.44 s), while Simple Averaging exhibited slightly higher latency (≈106.70 s), likely due to additional overhead in probability aggregation. Notably, the Weighted Averaging ensemble achieved both the highest accuracy and competitive inference time (≈98.86 s), suggesting that optimal weighting can provide accuracy gains without incurring substantial extra computational cost beyond other ensemble strategies.

Overall, Table 4 clearly supports the central hypothesis of this work: combining multiple pretrained CNNs through carefully designed ensemble fusion substantially improves classification performance over any single model, particularly for visually similar LULC categories.

5.3 Practical Implications and Deployment Considerations

The experimental results highlight clear trade-offs between accuracy, speed, and memory, which are essential when selecting a suitable LULC classification model for real deployment. Each architecture offers different strengths. MobileNetV2 remains the best option for lightweight or real-time scenarios, such as UAV monitoring or field devices, because it delivers fast inference and low memory use with acceptable accuracy. ResNet50 and EfficientNetB0 offer a balanced intermediate balance, combining strong accuracy with moderate computational demand, making them practical for routine operational workflows on standard GPU systems.

DenseNet121 provides the highest accuracy among single models but requires more processing time and memory, making it better suited for offline or server-based analysis where resource limits are less restrictive. For applications where reliability is critical—such as disaster

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

assessment, precision agriculture, or large-scale environmental monitoring—the weighted ensemble achieves the most dependable results. Its 99.43% accuracy represents a substantial reduction in errors compared with any single model, while the computational cost remains manageable for batch or scheduled processing. A selective or hybrid approach, where a fast model handles most predictions and the ensemble is used only for uncertain cases, can further improve efficiency without sacrificing accuracy.

Compared with earlier studies, which often rely on individual CNNs or small two-model ensembles, the proposed method offers a broader and more thorough evaluation. Previous work frequently emphasizes accuracy alone and rarely discusses inference time or memory requirements, limiting its usefulness for real-world deployment. Additionally, many studies overlook systematic comparisons of ensemble strategies, making it difficult to understand which fusion method provides the best trade-off.

By evaluating six pretrained CNN architectures and three ensemble fusion techniques within one consistent framework, this study provides a clearer picture of how diverse networks behave individually and when combined. The weighted ensemble not only surpasses previously reported performance on the UC Merced dataset but also offers practical guidance on when and how to use ensemble learning in operational LULC classification systems. Overall, the findings strengthen the evidence that combining complementary CNN architectures leads to more stable and accurate land-cover classification, especially in complex or visually similar categories.

6. Conclusion and Future Work

This study presented a detailed evaluation of ensemble learning for Land Use and Land Cover (LULC) classification using high-resolution remote sensing imagery. By combining six pretrained CNN architectures and systematically comparing three fusion strategies, the work demonstrated that ensembles significantly outperform individual models. The weighted averaging ensemble achieved **99.43% accuracy**, reducing errors by more than 80% compared with the best standalone network, confirming that diverse architectures offer complementary strengths that lead to more stable and reliable predictions.

The findings also highlight clear accuracy–efficiency trade-offs. Lightweight models such as MobileNetV2 are suitable for real-time and edge deployment, while ResNet50 and EfficientNetB0 offer a practical balance for routine operations. DenseNet121 remains the strongest single model for accuracy, whereas the weighted ensemble is the best choice for applications where reliability is critical, despite its higher computational cost. Together, these insights provide practical guidance for selecting models based on available resources and operational requirements.

Although results are strong, several limitations should be considered. The UC Merced dataset is small, balanced, and geographically limited, which may not reflect the diversity or imbalance of real-world imagery. The ensemble’s memory and inference requirements are also demanding, which can be challenging for large-scale or real-time deployment. In addition, the

Received: August 08, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

study focuses on single-date images; temporal dynamics, sensor variability, and interpretability concerns were not addressed.

Future Work

Future research should extend evaluation to larger and more diverse datasets, including multispectral, hyperspectral, and multi-temporal imagery. Developing efficient ensemble variants—such as knowledge distillation, dynamic model selection, early-exit strategies, or compressed architectures—would improve deployability. More advanced fusion approaches, including stacking, attention-driven combination, and uncertainty-aware weighting, may further enhance accuracy. Finally, integrating multi-modal data (optical, SAR, LiDAR) and exploring transformer-based or hybrid temporal models could broaden the applicability of ensemble learning for environmental monitoring and land-use change analysis.

Overall, this study shows that thoughtfully designed ensembles can deliver near-perfect LULC classification performance and offer a strong foundation for future advances in operational Earth observation systems.

References

1. Aldahoul, N., Abu Bakar, S. A. R., Jalal, A., & Ibrahim, A. (2023). *A systematic review of deep learning data augmentation techniques for remote sensing image analysis*. Remote Sensing Applications: Society and Environment, 29, 100887. https://doi.org/10.1016/j.rsase.2023.100887
2. Belgiu, M., & Drăguț, L. (2016). *Random forest in remote sensing: A review of applications and future directions*. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *ImageNet: A large-scale hierarchical image database*. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 248–255). https://doi.org/10.1109/CVPR.2009.5206848
4. Dietterich, T. G. (2000). *Ensemble methods in machine learning*. In T. G. Dietterich (Ed.), Multiple Classifier Systems (pp. 1–15). Springer. https://doi.org/10.1007/3-540-45014-9_1
5. Dou, S., Wang, Y., & Xu, P. (2024). *Ensemble convolutional networks for high-resolution land-use classification*. Remote Sensing, 16(2), 356. https://doi.org/10.3390/rs16020356
6. Ghosh, A., Joshi, P. K., & Oindrila, C. (2020). *A review of lightweight deep learning frameworks for land cover classification*. Earth Science Informatics, 13, 1121–1135. https://doi.org/10.1007/s12145-020-00488-0
7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90
8. Helber, P., Bischke, B., Dengel, A., & Borth, D. (2019). *EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(7), 2217–2226. https://doi.org/10.1109/JSTARS.2019.2918242

Received: August 08, 2025

9. Huang, G., Liu, Z., Van der Maaten, L., & Weinberger, K. Q. (2017). *Densely connected convolutional networks*. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 2261–2269). https://doi.org/10.1109/CVPR.2017.243
10. Khatami, R., Mountrakis, G., & Stehman, S. V. (2022). *The role of deep learning in land cover classification: A meta-analysis*. Remote Sensing of Environment, 268, 112760. https://doi.org/10.1016/j.rse.2021.112760
11. Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). *Deep learning in remote sensing: A comprehensive review*. ISPRS Journal of Photogrammetry and Remote Sensing, 152, 166–177. https://doi.org/10.1016/j.isprsjprs.2019.04.015
12. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). *MobileNetV2: Inverted residuals and linear bottlenecks*. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510–4520). https://doi.org/10.1109/CVPR.2018.00474
13. Shao, Z., Yang, D., & Yu, Z. (2019). Highly accurate land cover classification using ensemble learning. GIScience & Remote Sensing, 56(5), 701–716. https://doi.org/10.1080/15481603.2019.1577393
14. Shorten, C., & Khoshgoftaar, T. M. (2019). *A survey on image data augmentation for deep learning*. Journal of Big Data, 6(1), 60. https://doi.org/10.1186/s40537-019-0197-0
15. Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. International Conference on Learning Representations. https://arxiv.org/abs/1409.1556
16. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). *Rethinking the inception architecture for computer vision*. In IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818–2826). https://doi.org/10.1109/CVPR.2016.308
17. Tan, M., & Le, Q. (2019). *EfficientNet: Rethinking model scaling for convolutional neural networks*. In International Conference on Machine Learning (pp. 6105–6114). https://arxiv.org/abs/1905.11946
18. Yang, Y., & Newsam, S. (2010). *Bag-of-visual-words and spatial extensions for land-use classification*. In ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 270–279). https://doi.org/10.1145/1869790.1869829
19. Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., & Atkinson, P. M. (2019). *Joint deep learning for land cover and land use classification*. Remote Sensing of Environment, 221, 173–187. https://doi.org/10.1016/j.rse.2018.11.014
20. Zhao, W., Du, S., Zhang, L., & Mao, J. (2023). *Deep learning for high-resolution land use and land cover mapping: A review*. Remote Sensing, 15(5), 1221. https://doi.org/10.3390/rs15051221
21. Zhong, Y., Zhao, B., & Zhang, L. (2016). *A CNN ensemble framework for urban land-use classification using high-resolution remote sensing imagery*. Remote Sensing, 8(1), 1–22. https://doi.org/10.3390/rs8010001
22. Zhu, X. X., Tuia, D., Mou, L., et al. (2017). *Deep learning in remote sensing: A review*. IEEE Geoscience and Remote Sensing Magazine, 5(4), 8–36. https://doi.org/10.1109/MGRS.2017.2762307

Received: August 08, 2025