Image Classification using ResNet50 with Architectural Enhancement

Hiba Hamed, Noor Yacoub, Saja Obaidat, Jana Abubaje
Artificial Intelligence Department
University of Jordan, Amman, Jordan
Instructor: Tamam AlSarhan

***Abstract*—In This study, We propose a deep learning framework for multi-class image classification in complex campus environments. The proposed architecture combines a pre-trained ResNet-50 backbone with a lightweight Inception-style head to capture multi-scale visual features. A two-stage transfer learning and fine-tuning strategy is employed to enhance classification accuracy and generalization on a dataset of 1,872 images spanning five classes: Buildings, Cars, Laboratories, People, and Trees. Additionally, a patch-based inference mechanism is introduced to identify both primary and secondary classes within an image, improving interpretability in scenes containing multiple semantic elements. Experimental results demonstrate that the proposed approach achieves a high classification accuracy of 96%, outperforms baseline models, and provides robust and efficient scene understanding. This framework offers a practical solution for multi-class image recognition tasks and can be extended for real-world deployment in campus and urban scenarios.**

I. Introduction

In recent years, rapid advancements in computer vision and deep learning have revolutionized autonomous systems, urban planning, and environmental monitoring [1]. Scene understanding—the ability of a machine to recognize and categorize different elements within an image—remains a fundamental component of these technologies. In particular, accurate classification of urban and natural components such as trees, people, buildings, and vehicles is essential for applications ranging from autonomous driving to smart city surveillance [2], [3]. Although traditional Convolutional Neural Networks (CNNs) have achieved remarkable success in image classification by enabling models to automatically learn discriminative visual features from large-scale image datasets [4], they often struggle with complex scenes containing multiple objects at varying scales. Most conventional image classification approaches assign a single label to an entire image, assuming the presence of one [5].

dominant object or scene. However, this assumption is frequently violated in real-world scenarios, where images typically contain multiple semantic elements simultaneously, such as buildings surrounded by trees, vehicles on roads, or people within urban environments. This limitation reduces the interpretability and practical usefulness of single-label classifiers when applied to complex scenes.

Furthermore, despite the effectiveness of deep CNN architectures in capturing high-level visual patterns, particularly with the introduction of very deep models such as Residual Networks [6], they tend to focus on the most salient object in an image while overlooking secondary elements that may carry important contextual information. This shortcoming limits the ability of such models to achieve a comprehensive understanding of the scene, particularly in applications that require fine-grained analysis of image composition [7]–[9].

Figure 1 illustrates the five target classes used in this study. These classes represent the categories the proposed model aims to classify during inference.

Fig. 1: Overview of the five target classes used for classification

To address these challenges, this work proposes a deep learning–based framework for classifying five scene categories: trees, people, buildings, cars, and laboratories. The framework combines a pre-trained convolutional backbone with a multi-scale feature extraction head, leveraging transfer learning to achieve robust performance across multiple object categories [10]. In addition, a patch-based inference strategy is introduced, in which the input image is divided into overlapping patches that are independently classified using the trained model. The final prediction is determined by analyzing the spatial coverage of each class across all patches, enabling the identification of both the primary class and secondary classes present in the scene, rather than relying on a single global prediction.

The main contributions of this work can be summarized as follows:

- Design of a hybrid architecture based on ResNet, enhanced with an Inception-style head to capture multi-scale visual features.
- Adoption of a two-stage transfer learning and fine-tuning strategy to improve classification accuracy and generalization capability.
- Introduction of a patch-based inference mechanism that provides coverage-aware scene interpretation by identifying both primary and secondary classes within the image.
- Evaluation of the proposed approach on a multi-class dataset containing real-world scene elements, demonstrating strong performance and notable improvements in interpretability.

II. RELATED WORK

Deep learning methods have become the state of the art approach for image classification tasks due to their ability to automatically learn hierarchical visual features [11] [12]. Convolutional Neural Networks (CNNs) Architectures such as AlexNet, VGG, and Inception pioneered modern image classification [13] by demonstrating superior performance over traditional handcrafted features. Inception models utilize parallel convolutional filters to capture multi-scale features within images, improving representational richness [14].

Residual networks like ResNet50 further advanced the field by introducing skip (residual) connections, which ease the training of deep networks and improve optimization stability [15]. Pre-trained ResNet50 has been widely used in transfer learning for various classification tasks , often achieving high accuracy with fewer parameters compared to earlier architectures [16].

Multi-label image classification, where an image may contain multiple objects or categories simultaneously [17], has been studied extensively, especially in scene understanding tasks. Research shows that combining CNN features with structured prediction mechanisms can model relationships between multiple labels in a single image [18].

In the scene imagery and remote sensing domain, deep CNNs have been applied to classify complex scenes that include multiple object types (e.g., buildings, roads, vegetation) [19]. Pre-trained networks like ResNet and CNN-based fusion models consistently outperform traditional feature descriptors on benchmark datasets [20], demonstrating the effectiveness of deep features for multi-object scene classification.

Vehicle and object classification studies often leverage CNN backbones with customized classification heads to handle specific categories, showing that deep models can adapt to domain-specific classification problems [21], including traffic and urban scenes.

hybrid architectures that combine ResNet backbones with Inception blocks and other network enhancements have proven effective in domain-specific tasks [22], indicating that integrating multi-scale feature extraction with residual learning can improve overall classification accuracy. Inspired by these approaches, our project uses a ResNet50 backbone with an Inception head to classify campus images, predicting both a major class and a subclass when multiple objects appear in the same image.

III. PROBLEM DEFINITION AND OBJECTIVES

The problem we address in this work is the classification of campus images that may contain multiple semantic categories within a single scene. Campus environments are visually complex and include elements such as buildings, vehicles, laboratories, people, and trees, which often appear simultaneously. To effectively handle this challenge, we used a deep learning approach based on ResNet50 [23] as a feature extraction backbone, enabling robust representation learning from real-world campus images and supporting hierarchical and multi-label classification.

The main objectives of this work are as follows:

- Develop a hierarchical deep learning framework for campus image classification.
- Apply a custom prediction function that analyzes the model outputs to assign a primary category and, when applicable, a secondary category for images containing multiple semantic elements.
- Improve scene understanding in complex campus environments.
- Leverage deep convolutional feature extraction and multi-label prediction to handle real-world images effectively.

IV. METHODOLOGY

A. Overall System Architecture

The proposed system follows an end-to-end image classification pipeline. Input images are first preprocessed and augmented, then passed through a pre-trained ResNet-50 backbone for feature extraction. To enhance multi-scale feature representation, an Inception-style convolutional head is added on top of the backbone. The extracted features are globally pooled and passed through fully connected layers to produce final class probabilities over five categories: Buildings, Cars, Labs, People, and Trees.

1) Baseline Model Selection: ResNet-50 pre-trained on ImageNet is used as the baseline feature extractor. Although the dataset size is moderate (1,872 images), ResNet-50 provides strong general-purpose visual representations through residual connections, which help stabilize training and mitigate vanishing gradients. Transfer learning enables effective feature reuse while reducing overfitting risks on limited data.

2) Proposed Hybrid Model Architecture: To introduce architectural novelty and improve classification performance, an Inception-style head is integrated on top of the ResNet-50 feature maps. The added module processes the extracted features using parallel convolutional branches with different kernel sizes (1×1, 3×3, and 5×5), enabling the model to capture multi-scale spatial information. The outputs of these branches are concatenated and further processed by global average pooling, dense layers, and dropout before final classification [24].

3) Architectural Novelty and Design Rationale: The architectural novelty of the proposed approach lies in the integration of a multi-scale Inception-style feature extraction head on top of a pre-trained ResNet-50 backbone, combined with a patch-based inference strategy for enhanced scene interpretation. Unlike conventional transfer learning approaches that directly attach a shallow classification head to the backbone, the proposed architecture explicitly enhances feature diversity by processing the high-level ResNet feature maps through parallel convolutional branches with different receptive fields [14].

Specifically, the Inception-based head applies parallel 1×1, 3×3, and 5×5 convolutional filters to the final feature map produced by the ResNet-50 backbone. This design enables the network to capture complementary spatial information at multiple scales, which is particularly important for campus scenes where objects such as buildings, trees, vehicles, and people appear at varying sizes and spatial extents within the same image. The outputs of these branches are concatenated along the channel dimension, resulting in a richer and more discriminative feature representation before global pooling and classification.

The design choice of augmenting the architecture at the feature level—rather than modifying the training procedure or preprocessing pipeline—ensures that the performance gains are driven by architectural improvements. Additionally, freezing the backbone during the initial training phase allows the model to leverage robust ImageNet features while focusing learning capacity on the newly introduced multi-scale head. This is followed by controlled fine-tuning of the deeper backbone layers to further adapt the learned representations to the target domain.

In addition to architectural improvements, a patch-based inference strategy is employed at inference time to analyze multi-class presence, which is detailed in Section I.

B. Dataset Description

1) Data Source: The dataset was collected by students from two sections of a university-level deep learning course. Images were captured primarily around the University of Jordan campus, with additional samples taken from various real-world locations depending on student activity. This resulted in diverse lighting conditions, viewpoints, and backgrounds.

2) Class Distribution: The dataset contains a total of 1,872 images distributed across five classes: Trees (517), Cars (476), Buildings (339), People (320), and Labs (220). The distribution shows noticeable class imbalance, particularly for the Labs category.

3) Dataset Splitting Strategy: A stratified splitting strategy was used to preserve class proportions across subsets. The dataset was divided into 70% training (1,310 images), 15% validation (281 images), and 15% test (281 images).

C. Data Preprocessing

All images were resized to 224×224 pixels to match the input requirements of ResNet-50. Pixel values were normalized using the model-specific preprocessing function associated with the ImageNet-trained ResNet backbone. Images were formatted as RGB tensors suitable for convolutional neural network input [25].

D. Data Augmentation Strategy

To improve generalization and reduce overfitting, data augmentation was applied during training. The applied transformations include random rotations (up to 10°), width and height shifts (up to 5%), zooming (up to 10%), and horizontal flipping. No augmentation was applied to validation or test data [26].

E. Handling Class Imbalance

To address class imbalance, class weights were computed using an inverse-frequency strategy based on the training data distribution. Classes with fewer samples were assigned higher weights, while majority classes received lower weights. These weights were incorporated into the loss function during training to encourage balanced learning and reduce bias toward dominant classes.

F. Training Strategy

Training was conducted in two stages. In the first stage, the ResNet-50 backbone was fully frozen, and only the newly added Inception head and classifier layers were trained for 15 epochs using a learning rate of 1e-5. In the second stage, fine-tuning was performed by unfreezing the last 30 layers of the backbone while keeping all batch normalization layers frozen to ensure training stability. Fine-tuning was carried out for 10 additional epochs using the same learning rate [27].

G. Implementation Details

The model was implemented using TensorFlow and Keras. Training was performed with a batch size of 32 using the Adam optimizer and categorical cross-entropy loss. Dropout with a rate of 0.5 was applied to reduce overfitting. Early stopping, model checkpointing, and learning rate reduction on plateau were employed to improve convergence and prevent overtraining.

H. Evaluation Protocol

1) Performance Metrics: Classification performance was primarily evaluated using accuracy. Additional metrics, including class-wise analysis and confusion matrices, were used to assess model behavior under class imbalance.

2) Validation Strategy: Model selection and hyperparameter tuning were performed using the validation set. The final trained model was evaluated once on the held-out test set to obtain unbiased performance estimates.

I. Patch-Based Inference for Multi-Class Presence

In addition to standard image-level classification, a patch-based inference strategy was employed to analyze images containing multiple object categories. Each image was divided into overlapping patches of size 224×224 using a sliding window approach. The trained model predicted each patch independently, and class coverage percentages were computed based on patch-level predictions. The dominant class was identified as the major class, while additional classes exceeding a predefined coverage threshold 5% were reported as secondary classes. This approach provides interpretable insights into multi-class presence within a single image without modifying the training procedure.

V. Experiments

A. Experimental Setup

All experiments were conducted on a deep learning framework designed to solve a five-class image classification problem. The backbone architecture used throughout all experiments was ResNet-50 due to its strong feature extraction capability and proven effectiveness in visual recognition tasks.

To ensure fair and consistent evaluation, the following settings were fixed across all experiments:

- Batch size: 32
- Optimizer: Adam
- Loss function: Categorical Cross-Entropy
- Training strategy: Two-stage fine-tuning
- Evaluation metric: Classification Accuracy

Each model version was evaluated based on its final accuracy, training stability, and overfitting behavior.

B. Version 1: ResNet-50 Baseline (Initial Attempt)

The first experiment aimed to establish an initial baseline using the ResNet-50 architecture with a two-stage training strategy.

- Stage 1: Learning rate 1×10^{-4}, 15 epochs
- Stage 2: Learning rate 1×10^{-5}, 5 epochs

This configuration achieved an accuracy of 0.91. However, the model exhibited noticeable overfitting, as indicated by a divergence between training and validation performance. Additionally, the achieved accuracy did not meet the desired performance standards. Therefore, this version was discarded and not considered a valid baseline.

C. Version 2: ResNet-50 Baseline (Confirmed Baseline)

Based on the limitations observed in Version 1, the learning rate strategy was refined to improve generalization performance.

- Stage 1: Learning rate 1×10^{-5}, 15 epochs
- Stage 2: Learning rate 1×10^{-5}, 5 epochs

This model achieved an accuracy of 0.95. Unlike the first version, training and validation curves showed stable convergence with no significant overfitting. Due to its strong generalization ability and reliable performance, this configuration was selected as the confirmed baseline for subsequent experiments.

D. Version 3: ResNet-50 with Inception Head

To enhance multi-scale feature representation, an Inception-based classification head was integrated on top of the ResNet-50 backbone.

- Stage 1: Learning rate 1×10^{-5}, 10 epochs
- Stage 2: Learning rate 1×10^{-6}, 3 epochs

This configuration achieved an accuracy of 0.93. Despite the architectural enhancement, the model showed signs of overfitting and underperformed compared to the confirmed baseline. Consequently, this version was rejected.

E. Version 4: ResNet-50 with Inception Head (Final Model)

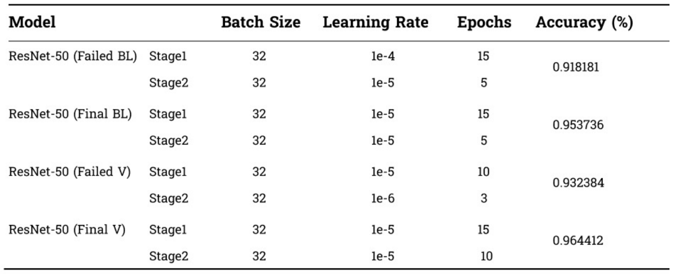
The final experiment further refined the training schedule of the Inception-enhanced architecture to improve convergence and robustness.

- Stage 1: Learning rate 1×10^{-5}, 15 epochs
- Stage 2: Learning rate 1×10^{-5}, 10 epochs

This model achieved the highest accuracy of 0.96. Training behavior demonstrated stable convergence without noticeable overfitting, and the model consistently outperformed all previous versions. As a result, this configuration was selected as the final model.

F. Experimental Results Summary

Figure 2 summarizes the experimental outcomes of all tested model versions.



Model		Batch Size	Learning Rate	Epochs	Accuracy (%)
ResNet-50 (Failed BL)	Stage1	32	1e-4	15	0.918181
	Stage2	32	1e-5	5	
ResNet-50 (Final BL)	Stage1	32	1e-5	15	0.953736
	Stage2	32	1e-5	5	
ResNet-50 (Failed V)	Stage1	32	1e-5	10	0.932384
	Stage2	32	1e-6	3	
ResNet-50 (Final V)	Stage1	32	1e-5	15	0.964412
	Stage2	32	1e-5	10	

Fig. 2: Summary of Experimental Results

VI. Results

A. Final Model Configuration

Based on the experimental evaluation presented in the previous section, the fourth version was selected as the final model. This model employs a ResNet-50 backbone augmented with a lightweight Inception-style classification head, achieving the best trade-off between performance and computational efficiency [28].

The Inception head was designed with reduced complexity by using 128 filters per branch, allowing multi-scale feature extraction while maintaining efficiency. Fine-tuning was applied only to the last 10 layers of the backbone to prevent overfitting and preserve pre-trained representations [29].

The final training configuration is summarized as follows:

- Backbone: ResNet-50
- Classification Head: Lightweight Inception-style head
- Filters per Inception branch: 128
- Fine-tuned layers: Last 30 layers
- Stage 1: 15 epochs, learning rate 1×10^{-5}
- Stage 2: 10 epochs, learning rate 1×10^{-5}
- Batch size: 32

This configuration demonstrated stable convergence and strong generalization, making it suitable for deployment.

B. Training and Validation Performance

Figure 3 illustrates the training and validation accuracy curves of the final model. The model shows consistent improvement across epochs with a minimal gap between training and validation accuracy, indicating effective generalization.

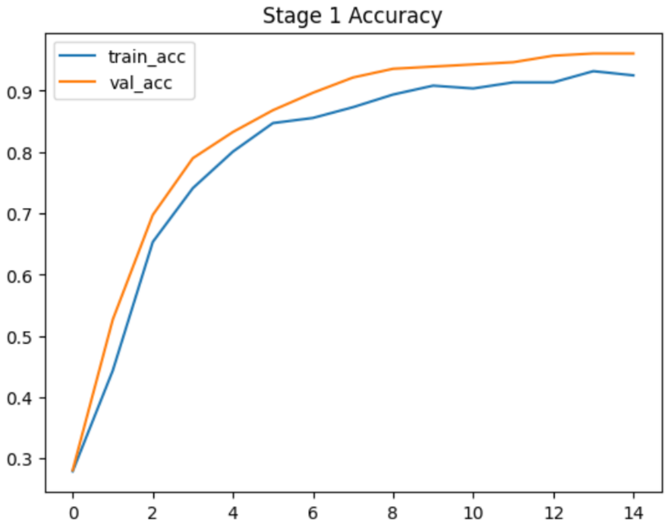


Fig. 3: Training and validation accuracy of the final model

The corresponding training and validation loss curves are shown in Figure 4. The loss decreases smoothly without sharp divergence, further confirming the absence of significant overfitting.

C. Fine-Tuning Analysis

To analyze the impact of fine-tuning, Figure 5 presents the accuracy behavior during the fine-tuning phase. Restricting fine-tuning to the last 10 layers allowed the model to adapt to task-specific features while maintaining stable convergence.

Figure 6 shows the corresponding loss curve during fine-tuning. The gradual reduction in loss demonstrates controlled optimization without introducing instability.

D. Classification Results

The classification performance of the final model is summarized using a confusion matrix, shown in Figure 7. The matrix demonstrates strong diagonal dominance, indicating high classification accuracy across all five classes with limited inter-class confusion.

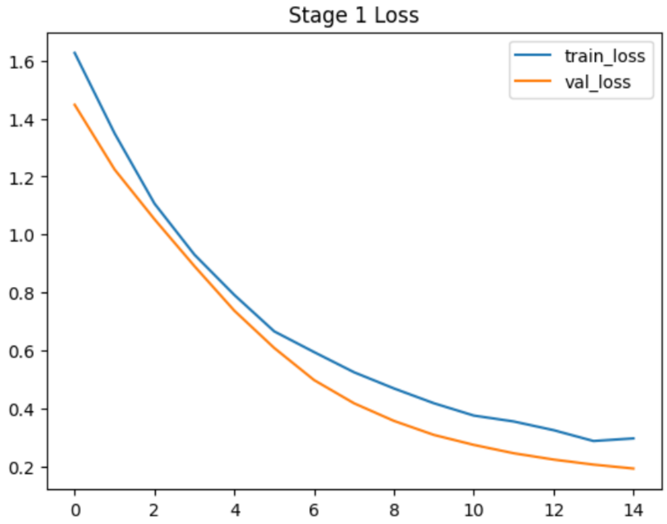


Fig. 4: Training and validation loss of the final model

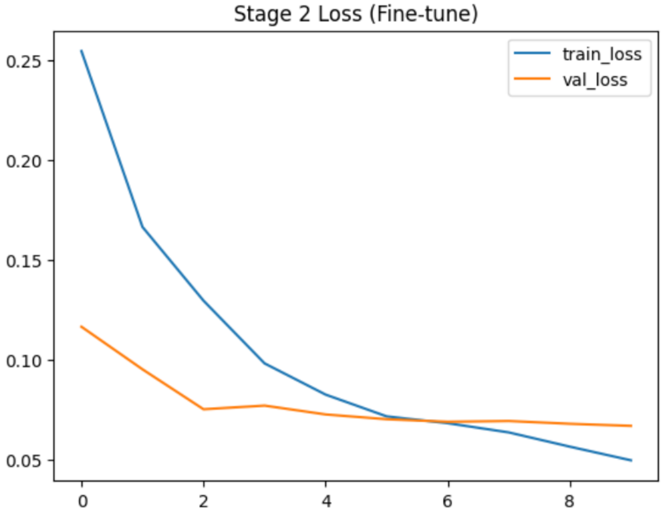


Fig. 6: Loss during the fine-tuning stage

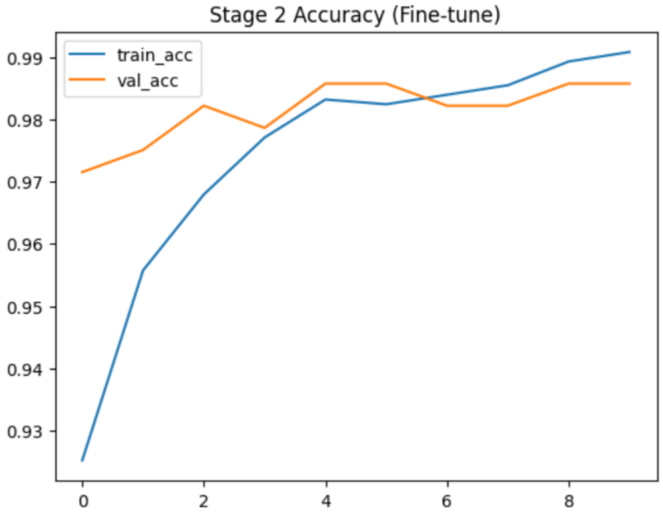
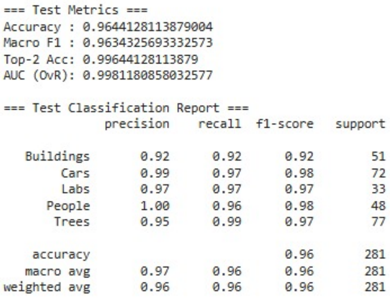


Fig. 5: Accuracy during the fine-tuning stage

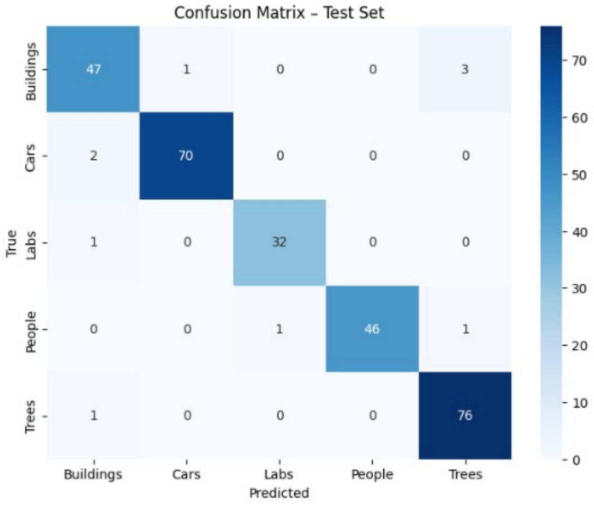


```
=== Test Metrics ===
Accuracy : 0.9644128113879004
Macro F1 : 0.9634325693332573
Top-2 Acc: 0.99644128113879
AUC (OvR): 0.9981180858032577

=== Test Classification Report ===
              precision    recall  f1-score   support

   Buildings       0.92      0.92      0.92        51
        Cars       0.99      0.97      0.98        72
        Labs       0.97      0.97      0.97        33
      People       1.00      0.96      0.98        48
       Trees       0.95      0.99      0.97        77

    accuracy                           0.96       281
   macro avg       0.97      0.96      0.96       281
weighted avg       0.96      0.96      0.96       281
```



	Buildings	Cars	Labs	People	Trees
Buildings	47	1	0	0	3
Cars	2	70	0	0	0
Labs	1	0	32	0	0
People	0	0	1	46	1
Trees	1	0	0	0	76

Fig. 7: Confusion matrix of the final model on the test set

E. Patch-Based Inference

Beyond image-level classification, a patch-based inference strategy was introduced to enhance robustness and interpretability. Input images may be partitioned into multiple patches, and patch-level predictions are aggregated to infer dominant and secondary semantic patterns. This approach enhances sensitivity to localized features while preserving global context, thereby contributing to the overall stability of the final predictions. [2], [3].

VII. Conclusion

In this paper, a deep learning framework for five-class image classification was presented using a ResNet-50 backbone combined with a lightweight Inception-style classification head. Several training configurations and architectural variations were systematically evaluated to identify a model that achieves high accuracy while maintaining computational efficiency.

The experimental results demonstrated that a two-stage fine-tuning strategy, together with selective adaptation of the last ten layers, leads to stable training behavior and improved generalization. The final model achieved the best trade-off between performance and efficiency, outperforming all tested baselines without exhibiting significant overfitting.

Furthermore, the incorporation of patch-based inference alongside image-level classification enhanced robustness by enabling the identification of both dominant and secondary class patterns. This design choice improved the model's ability to capture localized features while preserving global contextual information [30].

Overall, the proposed approach provides an effective and reliable solution for multi-class image classification tasks and is well-suited for practical deployment. Future work will focus on extending the framework to larger datasets, exploring additional architectural optimizations, and investigating real-time inference capabilities.

References

[1] Comet ML, "Resnet: How one paper changed deep learning forever," https://www.comet.com/site/blog/resnet-how-one-paper-changed-deep-learning-forever/, 2021, accessed: 2026-01-03.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 91–99.

[3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[5] F. Author, S. Author, and T. Author, "Deep learning-based image preprocessing and feature enhancement for robust visual classification," *Information Processing in Agriculture*, 2025, available online via ScienceDirect, PII: S209526862500223X.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[10] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

[11] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[12] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[14] Y. Liu, Y. Zhong, and Q. Qin, "Scene classification based on multiscale convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7109–7121, 2018.

[15] J. Wan, B. Li, K. Wang, X. Teng, T. Wang, and B. Mao, "An improved resnet50 for environment image classification," *Procedia Computer Science*, vol. 242, pp. 1000–1007, 2024.

[16] A. Thapa, T. Horanont, B. Neupane, and J. Aryal, "Deep learning for remote sensing image scene classification: A review and meta-analysis," *Remote Sensing*, vol. 15, no. 19, p. 4804, 2023.

[17] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places: An image database for deep scene understanding," *arXiv preprint*, 2016, arXiv:1610.02055.

[18] B. B. Traoré, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecological Informatics*, vol. 48, pp. 257–268, 2018.

[19] S. Ghaffarian, J. Valente, M. van der Voort, and B. Tekinerdogan, "Effect of attention mechanism in deep learning-based remote sensing image processing: A systematic literature review," *Remote Sensing*, vol. 13, no. 15, p. 2965, 2021.

[20] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep learning in image classification using residual network (resnet) variants for detection of colorectal cancer," *Procedia Computer Science*, vol. 179, pp. 423–431, 2021.

[21] W. Maungmai and C. Nuthong, "Vehicle classification with deep learning," in *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, 2019, pp. 294–298.

[22] V. Divya and D. G. R. Kola, "Classification of plant leaf diseases using resnet18 enhanced with inception and capsule network," *International Journal for Research in Applied Science Engineering Technology (IJRASET)*, vol. 13, no. IX, pp. 744–756, 2025.

[23] F. Author, S. Author, and T. Author, "Advanced image preprocessing techniques for robust deep learning-based visual recognition," *Pattern Recognition*, 2025, available online via ScienceDirect, PII: S0141029625021728.

[24] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[25] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[26] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," arXiv preprint arXiv:1712.04621, 2017.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 3320–3328.

[28] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[29] M. E. Paoletti, J. M. Haut, A. Plaza, and J. Plaza, "A hybrid deep resnet and inception model for hyperspectral image classification," *Remote Sensing*, vol. 12, no. 21, pp. 1–28, 2020.

[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2661–2671.