A Comparative Study of Different CNN Architectures for Real-World Image Classification in Bangladesh

Md. Sadmin Tahmid Khan
sadmid26tahmid@gmail.com

University of Dhaka

Ahad Bin Islam Shoeb
University of Dhaka

Arif Billah
University of Dhaka

Short Report

Keywords: Convolutional Neural Network, Image Classification, Vgg16, Resnet50

Posted Date: January 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8562599/v1

Additional Declarations: The authors declare no competing interests.

A Comparative Study of Different CNN Architectures for Real-World Image Classification in Bangladesh

Md. Sadmin Tahmid Khan[1,2*], Ahad Bin Islam Shoeb[2,3†] and Arif Billah[1,2†]

[1*]Computer Science and Engineering, University of Dhaka, Bangladesh.
[2]Computer Science and Engineering, University of Dhaka, Bangladesh.
[3]Computer Science and Engineering, University of Dhaka, Bangladesh.

*Corresponding author(s). E-mail(s): sadmid26tahmid@gmail.com;
Contributing authors: ahadbinislam1111@gmail.com;
arfbllh@gmail.com;
[†]These authors contributed equally to this work.

Abstract

Convolutional Neural Networks (CNNs) are widely used for image classification, yet their performance strongly depends on dataset complexity and deployment constraints. This study presents a comparative evaluation of custom-designed CNN architectures and popular pre-trained models on five real-world image datasets from Bangladesh, spanning agricultural and infrastructural applications. The tasks include mango variety classification (15 classes), paddy disease classification (35 classes), and three binary classification problems: road damage, footpath encroachment, and auto-rickshaw detection. In addition to task-specific CNNs, VGG16 and ResNet50 are evaluated using fixed feature extraction and transfer learning strategies. The results show that transfer learning, particularly with ResNet50, achieves the highest accuracy on complex multi-class datasets, while custom CNNs deliver competitive performance on binary tasks with substantially lower computational cost. These findings emphasize the trade-off between accuracy and efficiency and highlight the importance of selecting model architectures based on dataset characteristics and deployment requirements.

Keywords: Convolutional Neural Network, Image Classification, Vgg16, Resnet50

1 Introduction

Convolutional Neural Networks (CNNs) have become the dominant approach for image classification [1] across a wide range of application domains, including agricul-ture, infrastructure monitoring, and urban management. In recent years, large-scale CNN architectures pre-trained on benchmark datasets such as ImageNet [2] have been widely adopted due to their strong generalization capability and ability to learn rich hierarchical visual features. These pre-trained models can be used either directly as fixed feature extractors or adapted to new tasks through transfer learning, making them a popular choice for practical image classification problems.

In this work, the performance of widely used deep CNN architectures is evaluated on five distinct image classification tasks: Mango variety classification (15 classes), Paddy disease classification (35 classes), Road Damage detection (binary), Footpath Encroachment detection (binary) and Auto-rickshaw detection (binary). These tasks differ substantially in terms of class granularity, dataset size, visual complexity, and discriminative feature requirements. To systematically analyze model behavior across these diverse scenarios, two popular architectures—VGG16[3] and ResNet50 [4]—are employed under two different learning paradigms: (i) pre-trained feature extraction, where the convolutional base is kept frozen, and (ii) transfer learning, where selected layers are fine-tuned on the target dataset.

The objective of this study is to assess how well general-purpose pre-trained CNN models adapt to domain-specific classification tasks of varying complexity. By compar-ing VGG16 and ResNet50 under both pre-trained and transfer learning settings, this work aims to highlight the trade-offs between model depth, representational capacity, and adaptability to new domains. Experimental results across all four tasks provide insights into the effectiveness of transfer learning versus fixed feature extraction and illustrate how model choice impacts performance in real-world, application-driven datasets.

This analysis offers a practical evaluation framework for selecting suitable deep learning models for image classification tasks, particularly in applied settings where dataset characteristics and task requirements vary significantly.

2 Background Information

2.1 Convulational Neural Network

Convolutional Neural Networks (CNNs) are a class of deep neural networks specifically designed to process data with a grid-like topology[5], such as images. Unlike fully connected networks, where every neuron connects to all inputs, CNNs exploit spatial locality through convolutional operations, significantly reducing parameter count while preserving essential structural information. This architectural efficiency makes CNNs the dominant choice for image classification, detection, and recognition tasks across diverse domains, including agriculture, infrastructure monitoring, and urban analysis.

At the core of a CNN is the convolution layer, which applies learnable filters (kernels) across the spatial dimensions [6] of an input image as shown in Figure 1. Each filter extracts a distinct type of feature—such as edges, textures, or color gradients—by computing a weighted sum over local pixel neighborhoods. Earlier layers typically learn low-level visual cues (e.g., vertical or horizontal edges), while deeper layers capture increasingly abstract and semantic information (e.g., disease patterns on paddy leaves, shape outlines of mango varieties, or crack-like textures in road surfaces). Through backpropagation, the network automatically learns optimal filter values, replacing the need for handcrafted feature engineering.

2.2 Vgg16

VGG16 is a deep convolutional neural network architecture introduced by the Visual Geometry Group (VGG) at the University of Oxford[7]. It is characterized by its simple and uniform design, consisting of 16 weight layers arranged as a sequence of small 3×3 convolutional filters followed by max-pooling layers as shown in Figure 2. This con-sistent architecture allows VGG16 to learn highly expressive feature representations while maintaining structural simplicity.

Pre-trained on the ImageNet dataset[8], VGG16 has demonstrated strong perfor-mance in a wide range of image classification tasks. In practical applications, the convolutional layers of VGG16 can be used as a fixed feature extractor, leveraging learned representations such as edges, textures, and object parts. Despite its effec-tiveness, VGG16 is computationally intensive due to its large number of parameters, making it a useful benchmark model for analyzing performance versus complexity trade-offs.

2.3 Resnet50

ResNet50 is a deep residual network that addresses the degradation problem commonly observed in very deep neural networks. Introduced by He et al.[9], ResNet50 employs residual connections, also known as skip connections, which allow the network to learn residual mappings instead of direct transformations. These connections enable gradients to flow more effectively during training, allowing much deeper networks to be optimized reliably.

The ResNet50 architecture consists of 50 layers organized into residual blocks [10] that combine convolutional layers with identity shortcuts as shown in Figure 3. Pre-trained on ImageNet, ResNet50 captures robust and hierarchical feature represen-tations while maintaining better training stability than traditional deep CNNs. Due to its depth and residual structure, ResNet50 is particularly effective for complex visual tasks and often achieves higher accuracy with fewer training difficulties compared to earlier architectures.

2.4 Transfer Learning

Transfer learning is a training strategy in which a model pre-trained on a large-scale dataset is reused for a different but related task. Instead of training a CNN from scratch, the learned weights from a pre-trained network are leveraged to accelerate convergence and improve performance, especially when the target dataset is limited in size[11].

In image classification tasks, transfer learning is typically implemented in two ways. In the first approach, the pre-trained model is used as a fixed feature extractor, where the convolutional layers are frozen and only the final classification layers are trained on the target dataset [12]. In the second approach, known as fine-tuning, some or all of the pre-trained layers are unfrozen and retrained using task-specific data, allowing the model to adapt its learned representations to the new domain. Transfer learning is particularly effective in applied settings, as it balances computational efficiency with strong generalization performance across diverse classification tasks.

3 Design

The CNN architectures developed for this study were designed with careful consider-ation of the structure and visual complexity of the four Bangladeshi datasets. Rather than relying on large pre-trained models, the goal was to formulate lightweight, task-specific architectures that are computationally efficient, interpretable, and capable of extracting the specialized features required for each classification problem. Based on the number of classes and the level of fine-grained detail required, two distinct architectural families were created: a deeper five-block residual CNN for multi-class classification, and a comparatively simpler four-block CNN with spatial flattening for binary classification.

3.1 Multi-Class Classification Architecture

The multi-class datasets, consisting of fifteen mango varieties and thirty-five paddy disease classes, demand a high degree of discriminative power. The classes in these datasets often exhibit subtle visual differences, such as variations in texture, shape, or color patterns. To accommodate these demands, a progressive five-block CNN archi-tecture with residual connections was adopted. Each convolutional block increases the channel depth, allowing the network to build increasingly abstract levels of representa-tion. Residual connections were integrated into the deeper blocks to stabilize gradient flow, mitigate vanishing gradients, and enable the network to learn identity mappings when appropriate [13]. This improves convergence and allows the architecture to ben-efit from deeper hierarchical feature extraction without incurring excessive training instability.

Following the convolutional blocks, the network employs global average pool-ing instead of traditional flattening. Global average pooling significantly reduces the dimensionality of the feature maps, improving parameter efficiency and decreasing the risk of overfitting [14]. It also acts as a structural regularizer by enforcing translation invariance, which is particularly advantageous when samples vary in their spatial lay-out, as is common in field-collected mango and paddy images. The final classification stage consists of a dense layer with softmax activation, supported by progressively increasing dropout rates across blocks to enhance generalization. Despite its depth, this multi-class architecture maintains a compact parameter count of approximately eight million, demonstrating an effective balance between expressiveness and efficiency.

3.2 Binary Classification Architecture

The binary datasets, in contrast, require the network to identify the presence or absence of highly localized structural anomalies, such as cracks in road surfaces, phys-ical obstructions on footpaths and detection of auto-rickshaws. These tasks depend heavily on the spatial arrangement of visual cues, and therefore benefit from a design that preserves location-specific information. For this reason, a four-block CNN archi-tecture without residual connections was developed. The reduced number of blocks is suitable for the simpler decision boundaries involved in binary classification, while still providing sufficient feature extraction capability [15].

Unlike the multi-class model, the binary architecture replaces global average pool-ing with a flattening operation. Flattening maintains the complete spatial layout of the learned feature maps, ensuring that the fully connected classifier receives detailed positional information. This design facilitates the detection of small cracks, edges, and irregularities whose locations are critical to correct classification. Because flattening produces a high-dimensional input, the classifier is built from three fully connected layers, each contributing substantially to the model's parameter count. Indeed, more than ninety percent of the parameters in this model reside within the fully connected layers, resulting in a total of roughly twenty-six million trainable parameters. Although this is significantly larger than the multi-class architecture, it is justified by the need to preserve spatial fidelity during classification. Dropout is fixed at a constant rate of 0.5 to prevent overfitting in the dense layers, and sigmoid activation is used to produce the final binary output.

4 Methodology

The methodology of this study focuses on the creation, preparation, and evalua-tion of five custom CNN architectures designed for Bangladeshi image datasets. The approach integrates dataset-specific preprocessing, tailored model design, and stan-dardized training procedures to establish a fair and controlled comparison across tasks. This section outlines the datasets used, the preprocessing pipeline, the training configuration, and the evaluation strategy adopted throughout the experiments.

4.1 Datasets

The study evaluates performance on five distinct, real-world datasets originating from agricultural and infrastructural environments in Bangladesh. Each dataset presents different visual characteristics, class distributions, and levels of complexity, which directly influence architectural decisions.

The Mango Variety Dataset contains images of fifteen mango cultivars that show significant intra-class similarity and inter-class variation in terms of shape, texture, and coloration[16]. The Paddy Dataset includes thirty-five classes of paddy vari-ants, featuring visually subtle lesion patterns that require detailed hierarchical feature extraction[17]. In contrast, the Road Damage Dataset involves a binary classification task where images may or may not contain cracks or surface distortions, often local-ized within a small region of the frame[18]. The Footpath Encroachment Dataset, also binary, focuses on identifying the presence of obstacles or illegally placed objects that obstruct pedestrians, where both object type and spatial placement are essential[19]. Lastly, the Auto-Rickshaw Dataset involves identifying whether an auto-rickshaw is present in the image[20].

Across all datasets, image resolution, lighting conditions, and camera orientations vary, reflecting the uncontrolled nature of real-world data collection in outdoor Bangladeshi environments. These variations underscore the need for robust preprocessing and task-aligned CNN design.

4.2 Preprocessing and Data Augmentation

All images were resized to a standardized resolution of 224 × 224 pixels to ensure compatibility across architectures and to balance computational efficiency with sufficient detail preservation. Pixel values were normalized to a 0–1 range by dividing by 255, enabling stable gradient behavior during training.

To enhance generalization and mitigate overfitting—especially in datasets with limited samples—standard data augmentation techniques were applied. The augmentation strategy included random horizontal flips, rotations, zoom variations, and brightness adjustments. These transformations simulate natural variations in field conditions, such as changes in camera angle, shadows, and scale. Importantly, aug-mentations were applied only during training, ensuring that validation and test performance reflected the models' true generalization capability.

Class distributions were inspected for imbalance, particularly in the binary datasets. Where appropriate, uniform sampling strategies were used within batches to prevent bias toward majority classes.

4.3 Training Configuration

All models were trained under a unified configuration to enable controlled comparison across architectures. The Adam optimizer was selected due to its adaptive learning rate mechanism and proven stability in vision tasks. A learning rate of 0.001 was used consistently across all experiments.

A batch size of 32 offered an effective trade-off between gradient stability and GPU memory constraints. Training was performed for a maximum of 50–100 epochs depending on the dataset, with early stopping employed to prevent unnecessary computation once validation performance plateaued. Early stopping monitored validation accuracy with a patience threshold, terminating training when meaningful improvements ceased.

Regularization strategies varied slightly by architecture. Multi-class models utilized progressive dropout, increasing from 0.1 to 0.5 across deeper blocks to counteract the higher risk of overfitting associated with fine-grained classification. Binary models used a fixed dropout rate of 0.5 within the fully connected layers to regularize the high-capacity classifier created by flattening.

4.4 Implementation Details

All models were implemented in a modern deep learning framework (e.g., TensorFlow or PyTorch) with GPU acceleration to facilitate efficient training. The architectures followed the design principles outlined earlier, with consistent initialization methods and activation functions. Convolutional layers used ReLU activations and He initialization, while the final output layers used softmax for multi-class tasks and sigmoid for binary tasks.

Model checkpoints were saved based on the highest validation accuracy, ensuring that test results reflect the best-performing models rather than those at the final epoch. Training times varied according to model size, with multi-class models completing more quickly due to their lower parameter counts compared to the binary architectures.

4.5 Performance Metrics

To evaluate the effectiveness of the deep learning models across different classification tasks, multiple performance metrics are employed. Since the datasets vary in class distribution and task complexity (multi-class and binary classification), relying on a single metric may not provide a complete assessment of model performance. The selected metrics collectively measure classification correctness, class-wise prediction quality, and computational efficiency.

- Accuracy: Accuracy measures the proportion of correctly classified samples among the total number of samples. It provides an overall indication of model performance and is particularly informative when class distributions are relatively balanced.

- Precision: Precision quantifies the proportion of correctly predicted positive samples among all samples predicted as positive. It reflects the model's ability to minimize false positive predictions, which is important in applications where incorrect detections are costly.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: Recall, also known as sensitivity, measures the proportion of actual positive samples that are correctly identified by the model. It indicates how well the model minimizes false negatives, which is critical for tasks such as disease or damage detection.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1 Score: The F1 score is the harmonic mean of precision and recall, provid-ing a balanced evaluation of the model's classification performance. This metric is particularly useful for imbalanced datasets.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Computational Time: Computational time measures the efficiency of the model in terms of training and inference duration. It provides insight into the practical feasibility of deploying the model, especially when comparing pre-trained feature extraction and transfer learning approaches.

4.6 Evaluation Strategy

Evaluation was based primarily on classification accuracy, which measures the proportion of correctly predicted samples in the test set. Accuracy was selected as the primary metric due to its widespread use and its suitability for both multi-class and binary problems. For binary tasks, thresholding at 0.5 was applied to sigmoid outputs to determine class predictions.

To ensure reliable results, datasets were divided into training, validation, and test splits, maintaining consistent proportions across tasks. Validation accuracy was used for model selection during training, while test accuracy was reserved exclusively for final reporting.

Because the datasets represent practical applications in agriculture and infrastructure, accuracy directly reflects the models' ability to support real-world decision-making, such as identifying disease outbreaks or assessing road conditions.

5 Results

In this section , the performance of different deep learning models is analyzed across multiple datasets using a set of standard evaluation metrics. The objective is to com-pare the behavior of each model under varying dataset characteristics and to assess their effectiveness in both multi-class and binary classification tasks.

5.1 Mango Dataset

The evaluation begins with a detailed analysis of the Mango variety classification dataset. Figure 1 illustrates the variation of training accuracy and training error across epochs for the different CNN architectures, providing insight into their convergence behavior and learning stability. For this dataset, all performance metrics—Accuracy, Precision, Recall, F1 Score, and Computational Time—are examined to provide a comprehensive understanding of model performance.

Table 1 and Figure 2 presents a detailed comparison of model performance on the Mango variety classification dataset. The results show that transfer learning con-sistently improves classification performance compared to using pre-trained models as fixed feature extractors. Among all evaluated approaches, ResNet50 with transfer learning achieves the highest accuracy of 99.65%, along with superior precision, recall, and F1-score, indicating highly reliable and balanced class-wise predictions. This high-lights the effectiveness of deep residual architectures in capturing the subtle visual differences among mango varieties.

The VGG16-based models also demonstrate strong performance, with both the pre-trained and transfer learning variants achieving accuracies above 97%. While VGG16 with transfer learning slightly improves precision and recall compared to its pre-trained counterpart, it incurs a higher computational cost. In contrast, the custom CNN achieves competitive accuracy with significantly lower architectural complexity,

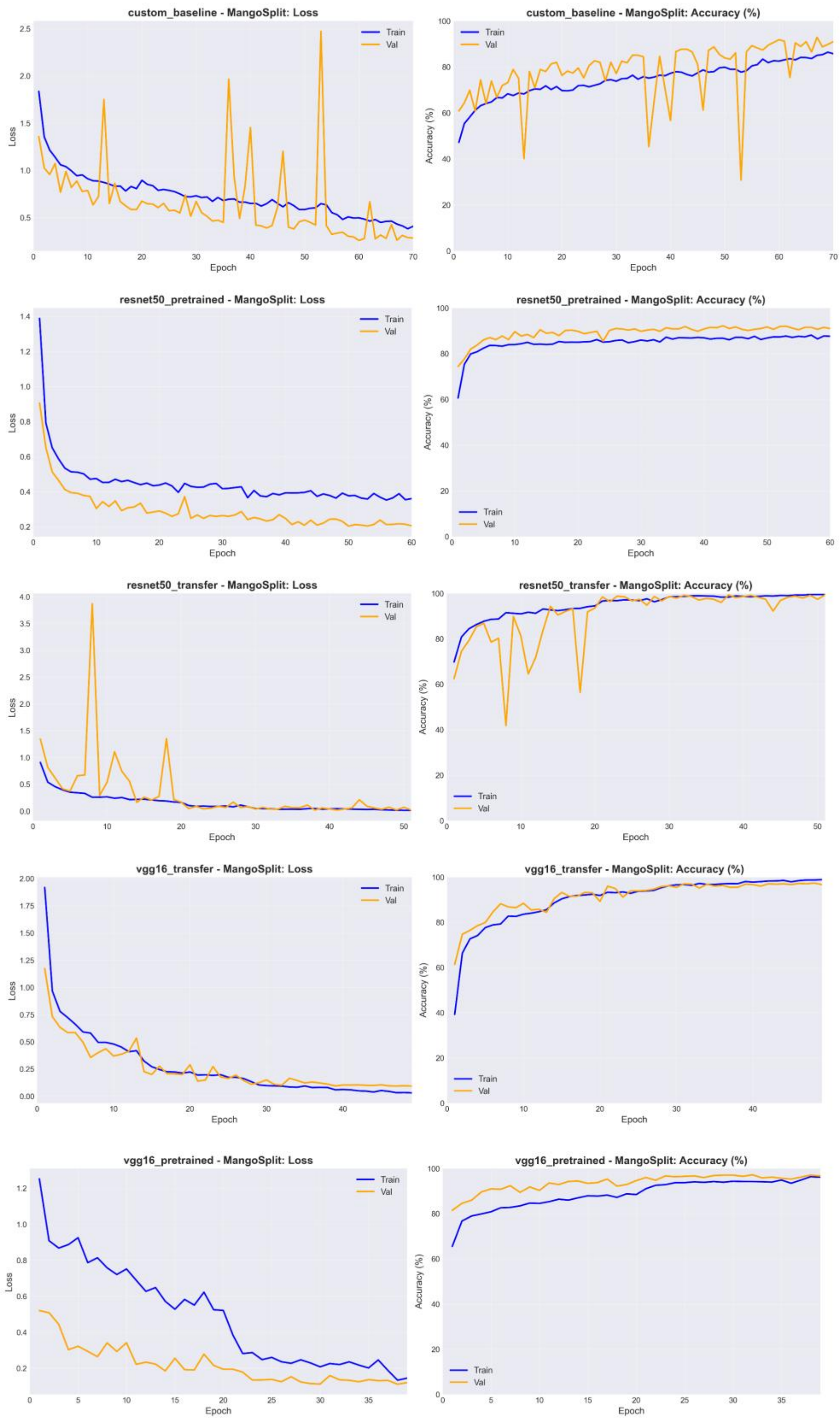

Fig. 1 Training error and accuracy variation for different CNN Architecture for Mango Dataset

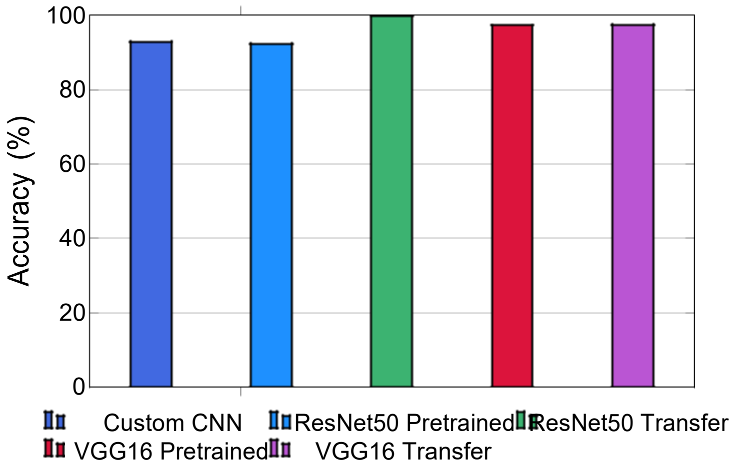

Fig. 2 Accuracy of different CNN Architecture for Mango Dataset

but lags behind deeper transfer learning models in fine-grained discrimination. Overall, these results confirm that ResNet50 with transfer learning provides the best trade-off between accuracy and robustness for the Mango dataset, particularly for complex multi-class classification tasks.

Among the evaluated models, ResNet50 with transfer learning demonstrates the best performance, achieving the highest classification accuracy on the Mango dataset.

Table 1 Performance Comparison of Different Models on Mango Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (s)
Custom CNN	92.91	90.71	90.14	89.64	2706
VGG16 Pretrained	97.23	96.26	95.77	95.74	1470
VGG16 Transfer Learning	97.40	96.35	96.30	96.29	3653
ResNet50 Pretrained	92.21	94.11	93.66	93.44	1877
ResNet50 Transfer Learning	99.65	98.62	98.59	98.56	2722

5.2 Rickshaw [1]

The evaluation next considers the Auto-Rickshaw detection dataset. Figure 3 shows the variation of training accuracy and training error across epochs for the different CNN architectures. Classification accuracy is used as the primary metric for com-parison, as summarized in Table 2 and Figure 4. Among all models, ResNet50 with transfer learning achieves the highest accuracy on the Rickshaw dataset, indicating its superior performance for this binary classification task.

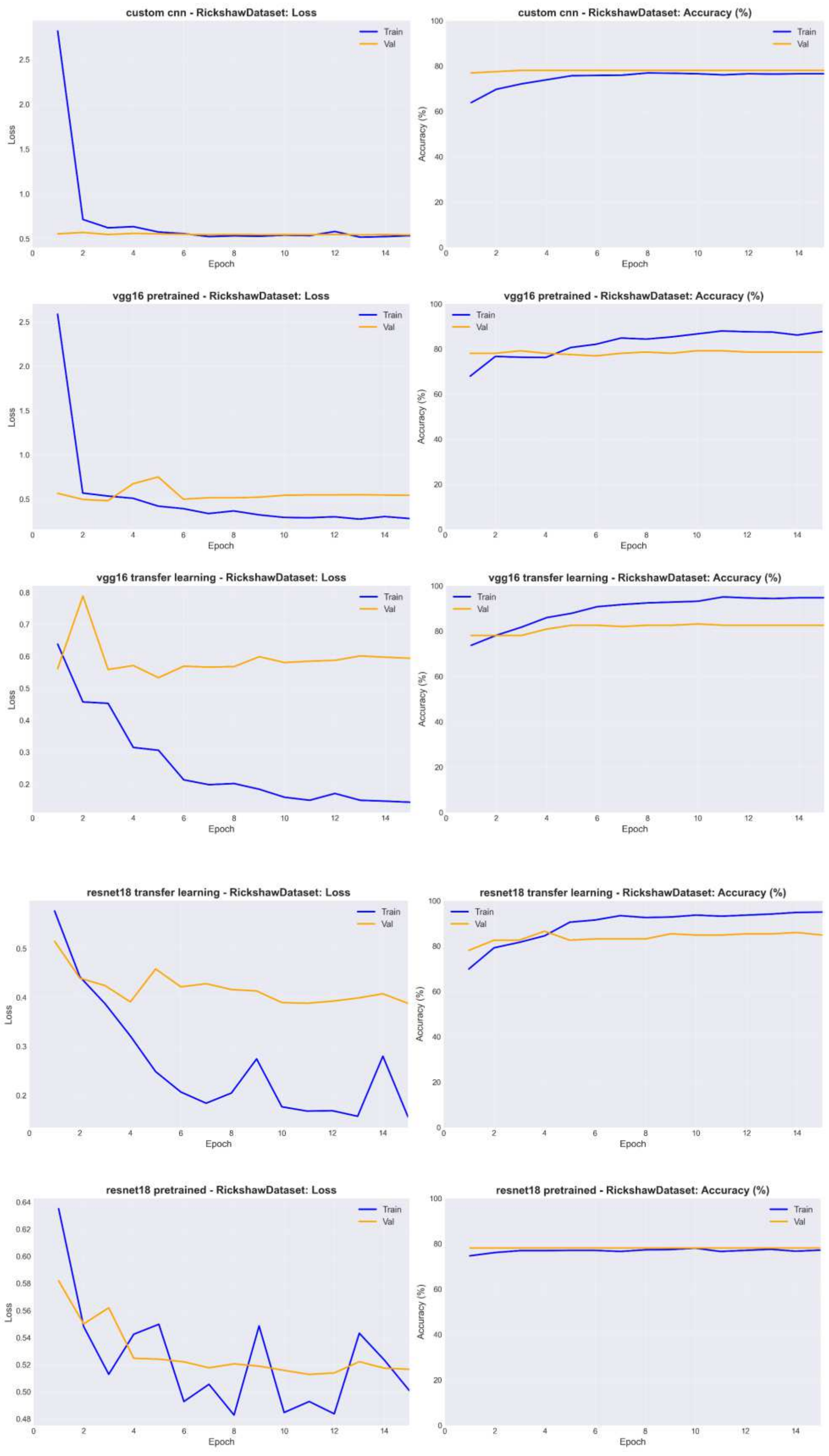

Fig. 3 Training error and accuracy variation for different CNN Architecture for Rickshaw Dataset

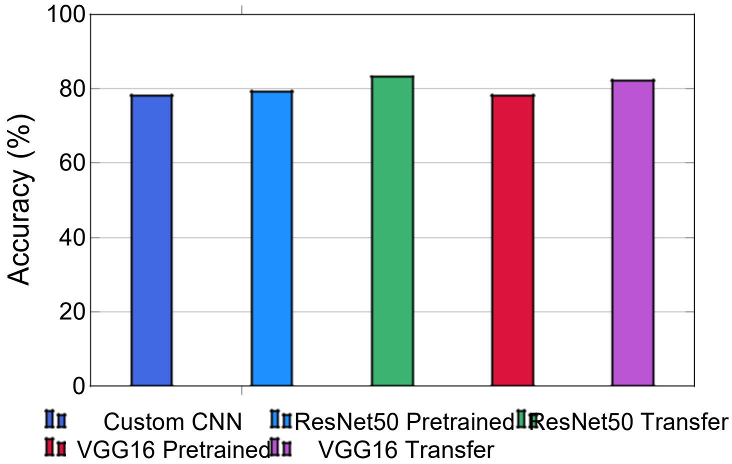

Fig. 4 Accuracy of different CNN Architecture for Rickshaw Dataset

Table 2 Performance Comparison of Different Models on Rickshaw Dataset

Model	Epochs	Accuracy (%)
Custom CNN	15	78.09
VGG16 Pretrained	15	78.53
VGG16 Transfer Learning	15	82.22
ResNet50 Pretrained	15	79.21
ResNet50 Transfer Learning	15	83.15

5.3 Paddy

The performance of different models is subsequently evaluated on the Paddy disease classification dataset. Figure 5 illustrates the variation of training accuracy and training error across epochs for the evaluated CNN architectures. The classification accuracy results are summarized in Table 3 and Figure 6.

ResNet50 with transfer learning outperforms the other models, achieving the high-est accuracy on the Paddy dataset, which highlights its effectiveness for complex multi-class classification tasks.

1

Table 3 Performance Comparison of Different Models on Paddy Dataset

Model	Epochs	Accuracy (%)	Time (s)
Custom CNN	166	87.29	15576
VGG16 Pretrained	80	74.71	6911
VGG16 Transfer Learning	90	75.20	10540
ResNet50 Pretrained	196	65.53	18691
ResNet50 Transfer Learning	76	93.36	17316

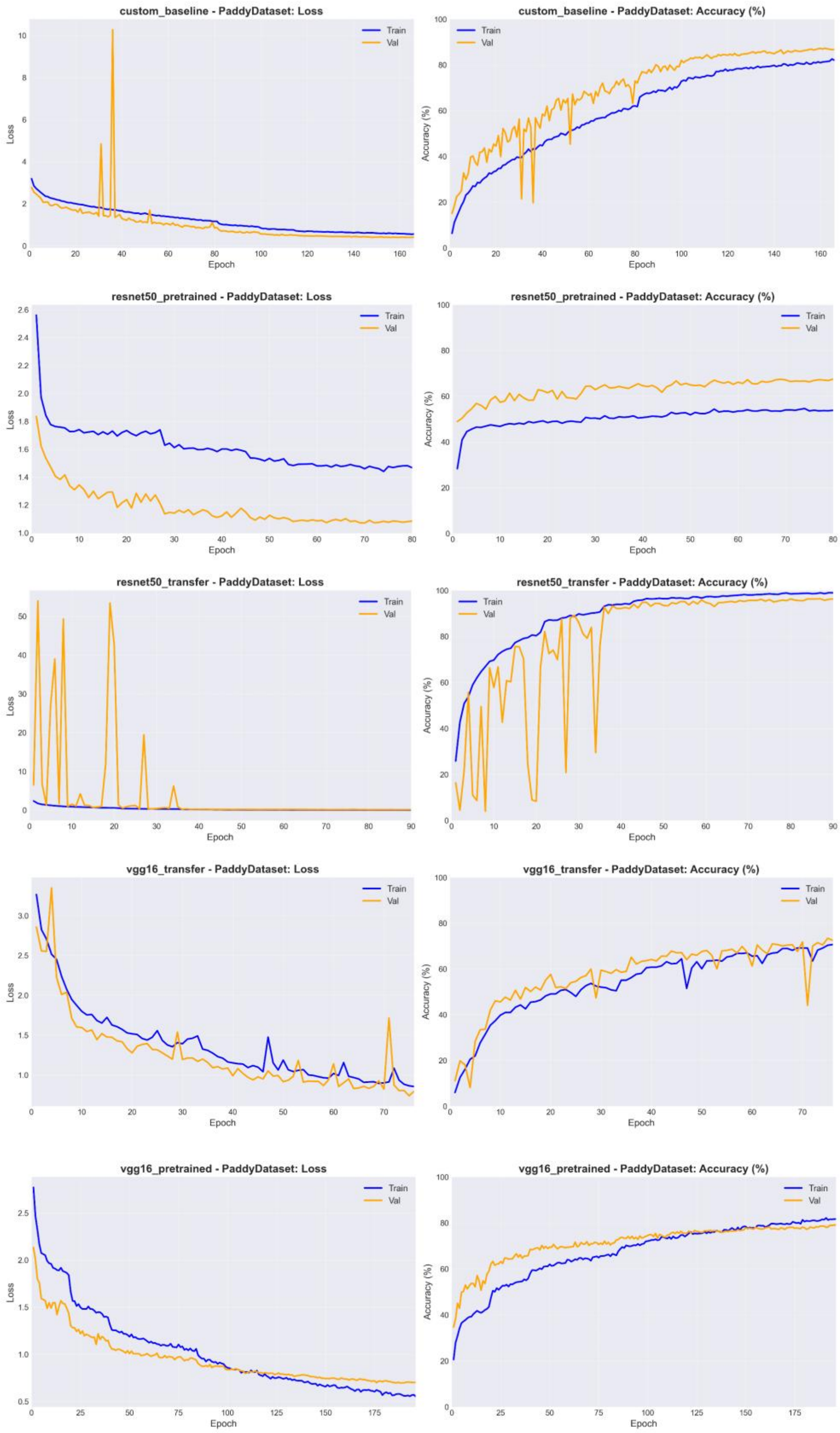

Fig. 5 Training error and accuracy variation for different CNN Architecture for Paddy Dataset

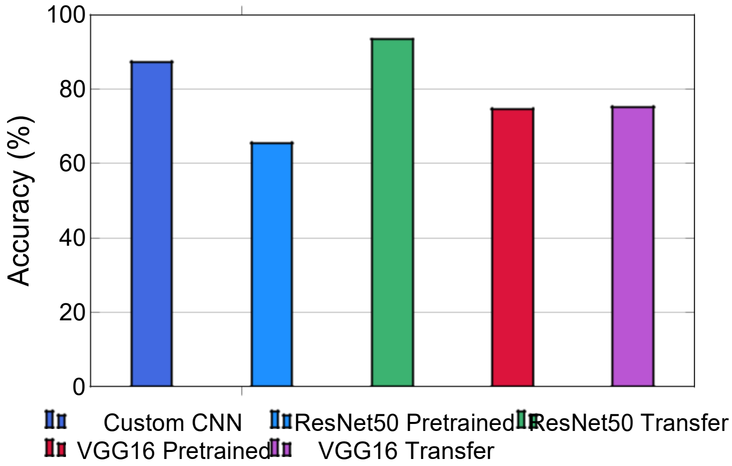

Fig. 6 Accuracy of different CNN Architecture for Paddy Dataset

5.4 Footpath

The evaluation then focuses on the Footpath Encroachment detection dataset. Figure 7 illustrates the variation of training accuracy and training error across epochs for the different CNN architectures, offering insight into their learning behavior and conver-gence characteristics. The classification accuracy results for all evaluated models are summarized in Table 4 and Figure 8.

Among the compared approaches, the pre-trained VGG16 model achieves the highest classification accuracy on the Footpath dataset, indicating that fixed fea-ture extraction is particularly effective for this task compared to deeper transfer learning–based models.

Table 4 Performance Comparison of Different Models on Footpath Dataset

Model	Epochs	Accuracy (%)
Custom CNN	50	87.84
VGG16 Pretrained	34	91.89
VGG16 Transfer Learning	58.11	89.19
ResNet50 Pretrained	19	87.94
ResNet50 Transfer Learning	12	89.19

5.5 Road Encroachment

The Road encroachment dataset is a binary classification task aimed at identifying the presence or absence of structural road damage in real-world images. These images often contain localized cracks, surface distortions, and irregular textures, making accu-rate detection highly dependent on the model's ability to capture both low-level edge

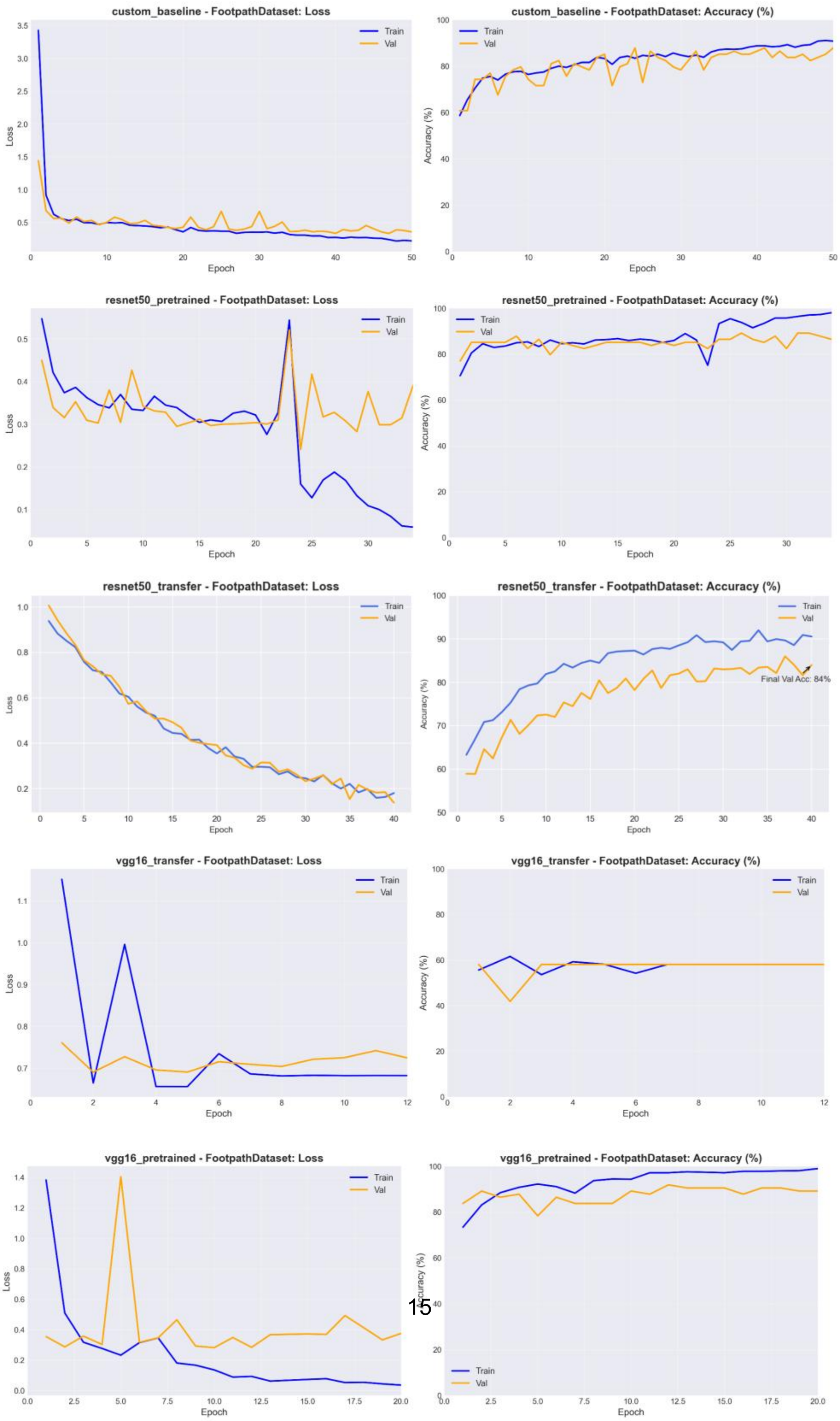

Fig. 7 Training error and accuracy variation for different CNN Architecture for Footpath Dataset

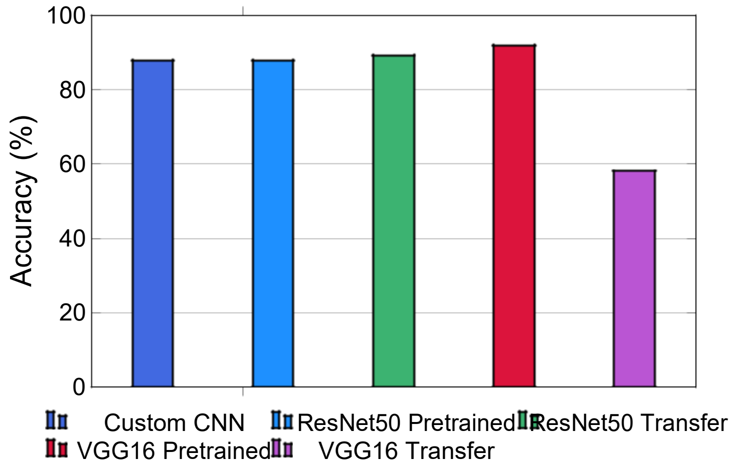

Fig. 8 Accuracy of different CNN Architecture for Footpath Dataset

information and higher-level spatial patterns. Figure 9 illustrates the variation of train-ing accuracy and training error across epochs for the evaluated CNN architectures, reflecting their learning stability during training.

The quantitative performance of all models on the Road dataset is presented in Table 5 and Figure 10. Among the evaluated approaches, ResNet50 with transfer learn-ing achieves the best overall performance, obtaining the highest accuracy (99.84%) along with consistently strong precision, recall, and F1-score values. This indicates its superior ability to generalize and reliably detect road encroachment patterns. VGG16 with transfer learning also performs competitively, achieving high scores across all metrics, while the pre-trained VGG16 model demonstrates strong performance with lower computational cost. In contrast, the custom CNN, although computationally efficient, yields comparatively lower accuracy, highlighting the advantage of deeper transfer learning–based architectures for this task.

Table 5 Performance Comparison of Different Models on Road Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Time (s)
Custom CNN	85.07	87.87	85.29	83.70	987
VGG16 Pretrained	97.06	97.31	97.06	97.09	1030
VGG16 Transfer Learning	98.53	98.60	98.53	98.54	1025
ResNet50 Pretrained	95.52	92.78	92.66	92.69	1056
ResNet50 Transfer Learning	99.84	98.62	98.59	98.56	958

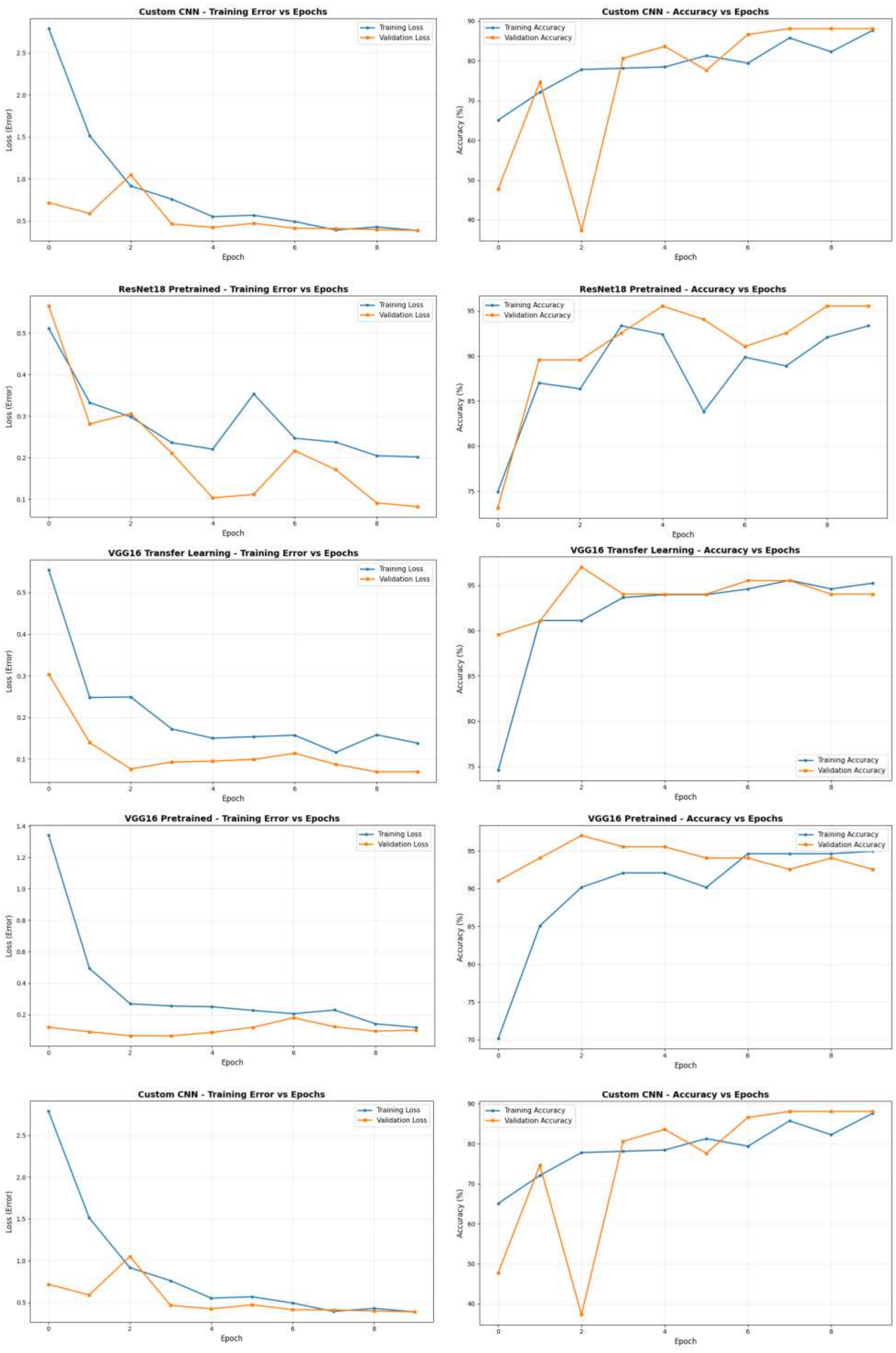

Fig. 9 Training error and accuracy variation for different CNN Architecture for Road Dataset

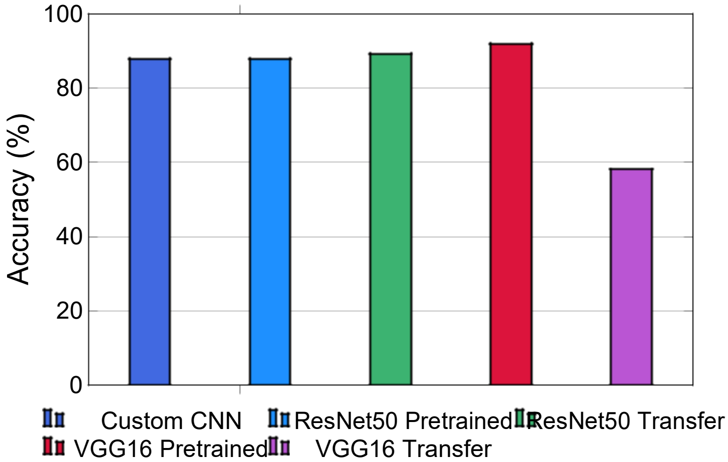

Fig. 10 Accuracy of different CNN Architecture for Road Dataset

6 Discussion

This section discusses the comparative performance of custom and pre-trained CNN models across multiple datasets, highlighting the trade-offs between classification accu-racy and computational efficiency, and identifying the most suitable approaches for different application scenarios.

The experimental results demonstrate clear performance differences among custom CNNs, pre-trained models, and transfer learning approaches across datasets of varying complexity. On the Mango dataset, which requires fine-grained discrimination among visually similar classes, transfer learning—particularly with ResNet50—achieves the best overall performance across all evaluation metrics. This indicates that deep residual architectures, when fine-tuned, are highly effective at adapting learned representa-tions to domain-specific agricultural imagery. In contrast, pre-trained models used as fixed feature extractors show reduced accuracy, highlighting the limitations of generic ImageNet features for specialized classification tasks.

Across datasets, transfer learning consistently improves accuracy, with the most significant gains observed in complex multi-class tasks such as Mango and Paddy clas-sification. For simpler binary tasks, including Road Damage, Footpath Encroachment, and Auto-Rickshaw detection, performance differences between models are smaller. In these cases, even the custom CNN achieves competitive accuracy, suggesting that

lightweight architectures can be sufficient when the visual patterns are more distinct and spatially localized.

A clear trade-off emerges between model complexity and practical feasibility. Custom CNNs are computationally efficient and require less memory, making them suitable for resource-constrained environments or real-time deployment. However, this efficiency comes at the cost of lower accuracy on fine-grained tasks. Conversely, pre-trained and transfer learning models offer superior accuracy and robustness but demand higher computational resources and longer training times, particularly for deeper architectures such as VGG16 and ResNet50.

Considering both performance metrics and resource constraints, transfer learning with ResNet50 is the most suitable approach for complex multi-class datasets where accuracy is critical, such as agricultural disease and variety classification. For binary infrastructure-related tasks or scenarios with limited computational resources, cus-tom CNNs provide a practical and efficient alternative. These findings emphasize the importance of selecting model architectures based on both dataset characteristics and deployment requirements.

7 Conclusion

This study evaluated custom CNN architectures and pre-trained deep learning models across five real-world image classification tasks from agricultural and infrastructural domains. The results show that transfer learning significantly improves performance for complex multi-class datasets, with ResNet50 consistently achieving the high-est accuracy and balanced precision–recall performance. In contrast, custom CNNs, while less accurate on fine-grained tasks, demonstrated competitive results on binary classification problems and offered clear advantages in computational efficiency.

These findings highlight the importance of selecting model architectures based on dataset complexity and deployment constraints. Deep transfer learning models are best suited for accuracy-critical applications, whereas lightweight custom CNNs provide a practical alternative for resource-constrained environments.

Future research will explore model compression and optimization techniques, such as pruning and quantization, to reduce the computational cost of deep transfer learn-ing models. Additionally, expanding the datasets and investigating attention-based architectures may further improve performance on fine-grained classification tasks.

References

[1] O’shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)

[2] Kieffer, B., Babaie, M., Kalra, S., Tizhoosh, H.R.: Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In: 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–6 (2017). IEEE

[3] Qassim, H., Verma, A., Feinzimer, D.: Compressed residual-vgg16 cnn model for big data places image recognition. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 169–175 (2018). IEEE

[4] Koonce, B.: Resnet 50. In: Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 63–72. Springer, ??? (2021)

[5] Wu, J.: Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China 5(23), 495 (2017)

[6] Ketkar, N., Moolayil, J.: Convolutional neural networks. In: Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch, pp. 197–242. Springer, ??? (2021)

[7] Srivastava, S., Kumar, P., Chaudhry, V., Singh, A.: Detection of ovarian cyst in ultrasound images using fine-tuned vgg-16 deep learning network. SN Computer Science 1(2), 81 (2020)

[8] He, K., Girshick, R., Doll´ar, P.: Rethinking imagenet pre-training. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4918–4927 (2019)

[9] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[10] Mascarenhas, S., Agarwal, M.: A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification. In: 2021 International Conference on Disruptive Technologies for Multi-disciplinary Research and Applications (CENTCON), vol. 1, pp. 96–99 (2021). IEEE

[11] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. Proceedings of the IEEE 109(1), 43–76 (2020)

[12] Shin, H.-C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mol-lura, D., Summers, R.M.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging 35(5), 1285–1298 (2016)

[13] Farooq, A., Anwar, S., Awais, M., Rehman, S.: A deep cnn based multi-class classification of alzheimer's disease using mri. In: 2017 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1–6 (2017). IEEE

[14] Iqbal, I., Younus, M., Walayat, K., Kakar, M.U., Ma, J.: Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images. Computerized medical imaging and graphics 88, 101843 (2021)

[15] Tufail, A.B., Ma, Y.-K., Zhang, Q.-N.: Binary classification of alzheimer's disease using smri imaging modality and deep learning. Journal of digital imaging 33(5), 1073–1090 (2020)

[16] Ferdaus, M.H., Prito, R.H., Ahmed, M., Islam, M.M., Ali, M.S., Ibrahim, M., Rasel, A.A.S., Islam, M., Jabid, T., Rahman, M.A., et al.: Mangoimagebd: An extensive mango image dataset for identification and classification of various mango varieties in bangladesh. Data in Brief, 111908 (2025)

[17] Tahsin, M., Ibrahim, M., Nafisa, A.T., Nuha, M.B.R., Arnab, M.I., Ferdaus, M.H., Islam, M.M., Rashid, M.R.A., Jabid, T., Ali, M.S., et al.: Paddyvarietybd: Clas-sifying paddy variations of bangladesh with a novel image dataset. Data in Brief 60, 111514 (2025)

[18] Hossen, R., Mistry, D., Rahman, M., Hridoy, W.A.S.A.R., Saha, S., Ibrahim, M.: Road damage and manhole detection using deep learning for smart cities: A polygonal annotation approach. arXiv preprint arXiv:2510.03797 (2025)

[19] Lubaina, A., Pahlowan, M.E.U., Munni, T.I., Ibrahim, M.: Footpathvision: A comprehensive image dataset and deep learning baselines for footpath encroach-ment detection. Available at SSRN 5577201 (2024)

[20] Sukanto, S.D., Roy, D., Shakil, F., Singha, N., Asik, A., Joarder, A., Fuad, M.M.N., Ibrahim, M.: Detecting unauthorized vehicles using deep learning for smart cities: A case study on bangladesh. arXiv preprint arXiv:2510.26154 (2025)