Comparative Study of Plant-Pathogen Diagnosis Using Custom Convolutional Neural Networks and Pretrained Deep Learning Models

Abstract— Plant diseases significant threat to global food security, causing crop yield losses every year. Early and accurate pathogen classification such as bacteria, fungi, viruses, pests, and healthy plants is crucial for effective disease management. This study proposes a custom Convolutional Neural Network (CNN) model that achieved 96.88% test accuracy in pathogen classification from leaf images. The custom CNN model compares with pretrained models like VGG16 (93.92%), MobileNetV2 (73.42%), and DenseNet121 (86.97%). The model used Batch Normalization, Dropout, and L2 regularization to enhance generalization and reduce overfitting in the real-world scenario. We used data augmentation such as rotation, zooming, flipping to improve model performances. We also used model performance metrics, including precision, recall, F1-score, ROC, AUC score, confusion matrix to understand the models evaluation.

Keywords— Plant Disease Classification, Convolutional Neural Network (CNN), Deep Learning, Pathogen Detection

Introduction

Plant disease is a significant global threat to agricultural production that negatively affects food security and economic stability. Several plant pathogens, such as bacteria, fungi, viruses, etc., are the leading causes of crop yield loss worldwide. Plant pathogens infect plants by invading plant tissues, and damaging cell structures with toxins and enzymes [1]. Pathogens damage plants by causing diseases like blights, necrotic spots in leaves even sometimes plant death, and due to pathogen damage total of 16% loss of global crop production [1]. In the early stage accurately identifying the disease which pathogen affected is essential for minimizing overall damage. In this case, primarily identifying the pathogen classification, such as bacteria, fungi, and viruses, is crucial for guiding treatment and the further stages.

With the advancement of deep learning, we can make the primary pathogen identification process easy by using image classification. In this study, we used Convolutional Neural Networks (CNNs). In the custom CNN architecture, we used conv2d, maxpooling2d along with batch normalization, dropout layers, and L2 regularization to prevent overfitting and improve generalization. Also make the dataset efficient and correctly identify the class in the real world scenario, we used data augmentation techniques such as rotation, zooming, and flipping. In this dataset, the total number of training images was 31997, and the testing and validation images were 4000 and 4000 [2].

In this study, our custom CNN model achieved better accuracy compared with other three pretrained models, including VGG16, MobileNetV2, and DenseNet121. Our custom achieved

96.88% test accuracy. Pretrained model VGG16 achieved 93.92%, MobileNetV2 achieved 73.42%, and DenseNet121 achieved 86.97% accuracy score. We also used various performance evaluation metrics such as precision, recall, F1-score, confusion matrix, ROC score and AUC scores. AUC scores confirmed the model's effectiveness across all five classes, each achieving an AUC of 1.00. Overall, this study demonstrates the effectiveness of practical plant disease classification

In this study our key contributions are—

I. A custom CNN model designed for pathogen classification.
II. Comparative analysis with pretrained models including VGG16, MobileNetV2, and DenseNet121.

Literature Review

Early detection and classification of pathogen-related plant diseases is crucial for crop production damage repair. There are several recent research works focused on plant-pathogen image classification by using computer vision techniques. Sladojevic et al. developed a plant disease detection system using 13 different plant diseases and a model trained with the Caffe deep learning framework that achieved a precision of 96.3% accuracy on average [3]. Sivabalaselvamani et al., in their proposed architecture, used image processing, fuzzy clustering, and segmentation techniques, and their proposed segmentation achieved 96.74% accuracy [4]. Negi et al., proposed a Deep CNN model trained on a large agricultural plant dataset to accurately detect and identify leaf diseases and they achieved 96.02% validation accuracy [5]. Haridasan et al., proposed system combines image processing, SVM, and CNN to detect and classify five rice crop diseases, and they achieved a validation accuracy of 91.45% [6]. Sinha et al., used GLCM-based texture features and impact-level classification technique to distinguish spot and blight diseases across four plant species, and they achieved 74% accuracy [7]. Wang et al., used deep CNN models including VGG16 on apple black rot images, and they achieved 90.4% accuracy in disease severity classification [8].

We also explore three different pretrained models to compare with our custom CNN model such as VGG16, MobileNetV2, and DenseNet121. The VGG16 paper shows that increasing CNN depth to 16-19 layers with small (3×3) convolution filters significantly improves accuracy and well performed ImageNet datasets [9]. MobileNetV2 improves mobile model efficiency with inverted residual blocks and linear bottlenecks, achieving strong performance in image classification [10].

Methodologies

In this section, we explore the dataset preprocessing, data augmentation, custom CNN model development, training and evaluation using performance metrics.

A. **Data Preprocessing and Augmentation**

In this study, we used a publicly available plant pathogen dataset [2]. In this dataset, there are five classes Bacterial, Fungal, Viral, Healthy, and Pest-infected. The dataset contains a total of 39997 images in five classes. In the data preprocessing process, we used 80% data for training, 10% for validation, and 10% data for testing purposes. All images were resized to 150×150 **pixels**. To make the dataset better generalization, we used several preprocessing and data augmentation techniques. The pixel values are normalized as 0 and 1 range. To overcome model overfitting and make the model predict correctly in real world space we use data augmentation where applied random rotations (±20°), horizontal and vertical shifts (20%), shear distortion (20%), zooming (20%), and horizontal flipping. The coding part was implemented using the

Fig. 1. Data Visualization

Kaggle environment with GPU P100 support, utilizing the TensorFlow Keras library for model development and training. Figure 1 shows the random images of all 5 classes.

B. Custom CNN Model Architecture

We developed deep custom Convolutional Neural Network (CNN) model optimized for multiclass classification of plant pathogens. The model was carefully structured to extract hierarchical spatial features to ensure better real-world performance. The input of the model was RGB images 150 X 150 X 3 and the output layer for five neurons represents 5 classes. In the figure 2, shows the custom CNN model architecture where used Conv2d, batch normalization, maxpooling2d, dropout, flatten, and dense layer. In the model there are 7 convolutional layers with filter sizes used 32, 64, 64, 128, 128, 256, and 512. Each of the convolutional layers used kernel size as 3 X 3, padding same, and activation function ReLU. Along with each of the convolutional layer used batch normalization and MaxPooling2D layer. In the MaxPooling2D layer used pooling size 2 X 2 to reduce dimensionality and control overfitting. We also used dropout layers with 0.25 dropout rate. Then, a flatten layer, 3 fully connected layers, and at the

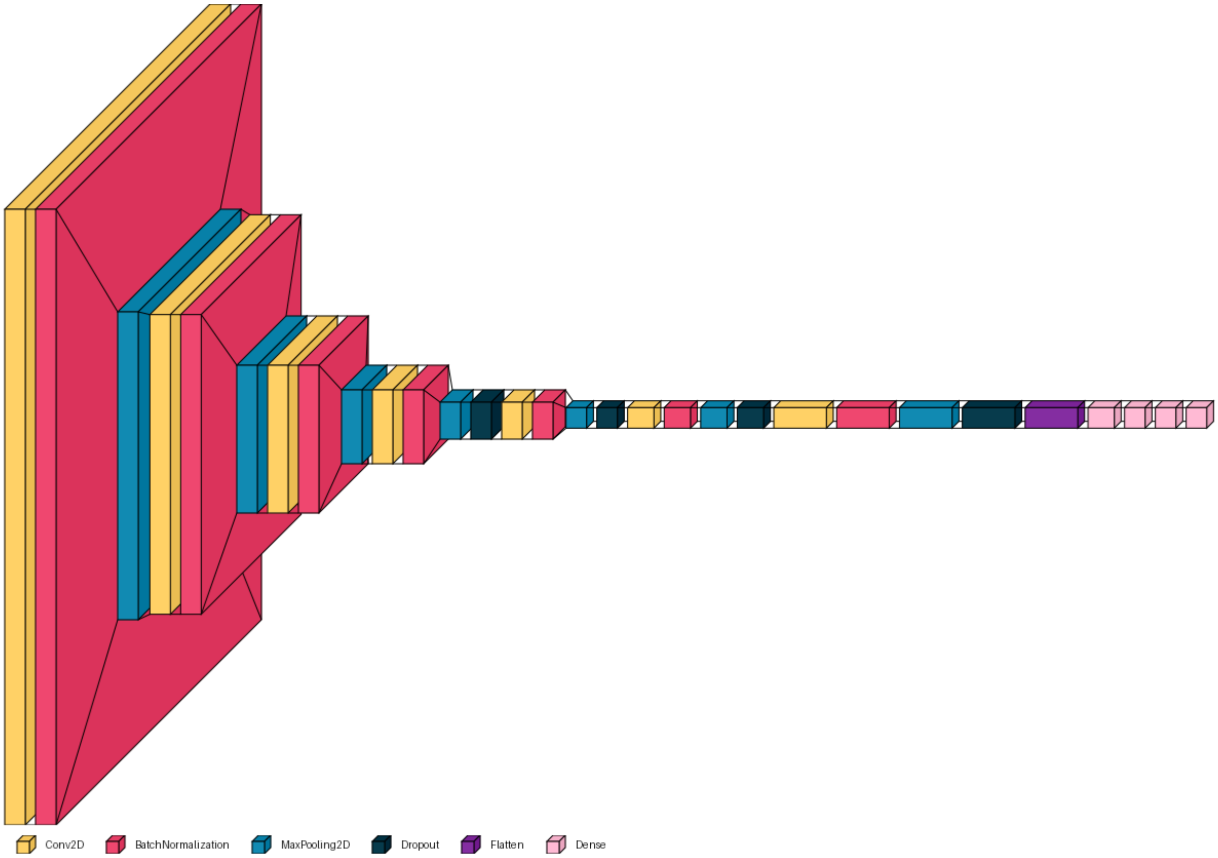


Fig. 2. Custom CNN Model Architecture

end output layer. Each of the fully connected layers used l2 regularization and 256, 128, 32 units. Finally, we used output layer for 5 classes.

C. **Evaluation Metrics**

To ensure the performance in our custom CNN model, we used multiple evaluations. The evaluation metrics include precision, recall, f1-score, confusion matrix, ROC curve, and AUC score.

$$Precision \frac{TP}{TP+FP} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

$$Recall = \frac{TP}{TP+FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Here equation 1 and 2 shows the equation of precision and recall, where TP means true positive, FP means false positive, FN means false negative.

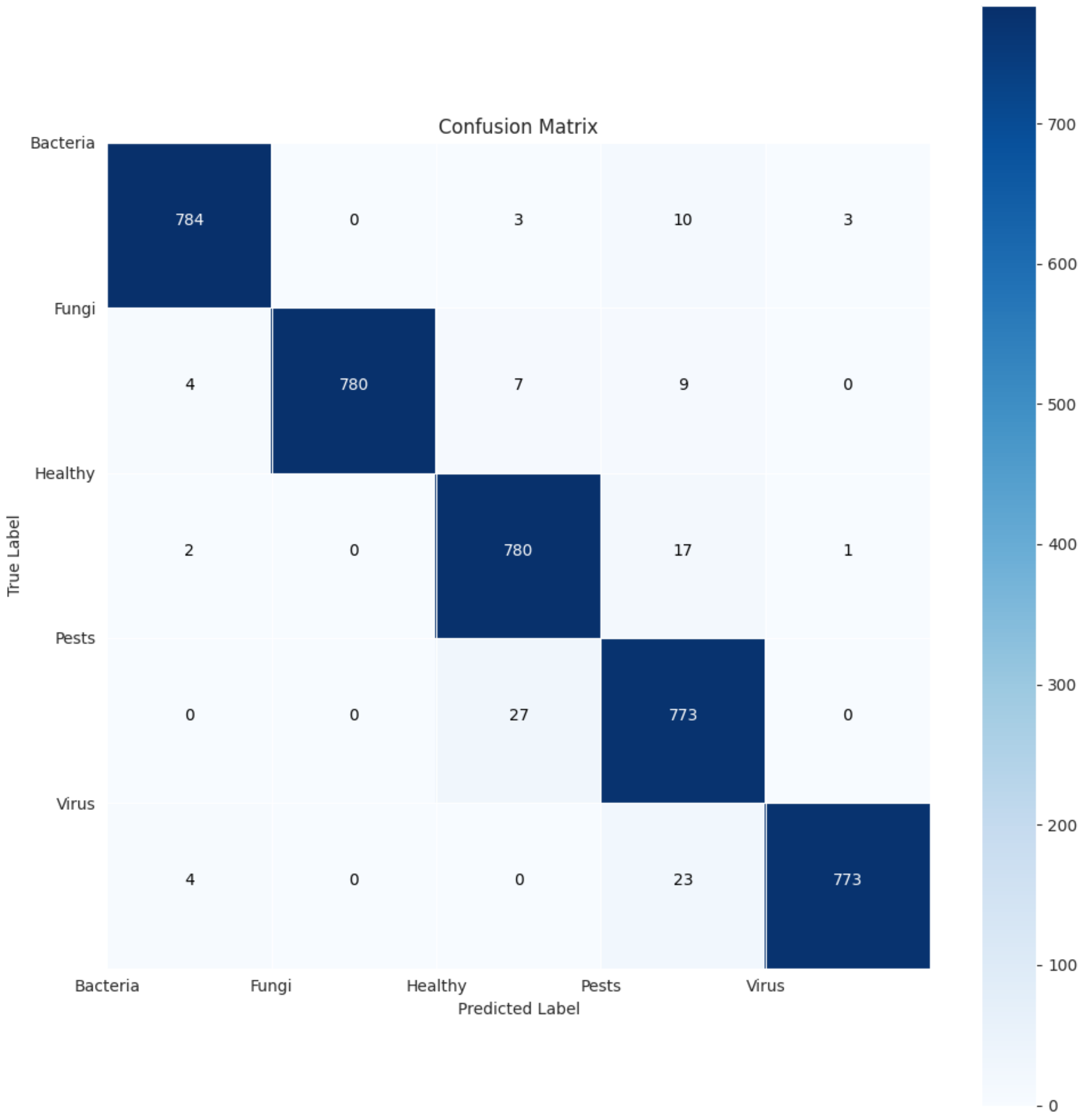
Experimental Results

We used 32 batch sizes, optimizer as Adamax with 0.001 learning rates, and categorical cross entropy as loss function. After that, train the model. The results confirm the effectiveness, robustness, and practicality of the proposed approach. First of all, in the classification report, the model achieved overall 96.88% test accuracy. The model also shows 97.25% train and 96.37% validation accuracy. In the table I is represent classification report, where the model shows strong classification capability across all classes. The model shows highest F1-score in the fungi class which is 99% and the lowest for the pests class which is 95%. Also the model shows precision score highest 100% for fungi 99% for bacteria, and 99% for viruses.

Table I: Classification Report

	Precision	Recall	F1-score
Bacteria	0.99	0.98	0.98
Fungi	1.00	0.97	0.99
Healthy	0.95	0.97	0.96
Pests	0.93	0.97	0.95
Virus	0.99	0.97	0.98
accuracy			0.97

Figure 3, shows the confusion matrix provides models predictive performances of all 5 classes. The diagonal elements indicate correct predictions each of the class. It shows that, bacteria class correctly identifies 784 out of 800 and 16 shows minor misclassifications. Also correctly identify 780 in fungi and healthy and 773 correctly identify in pests and virus classes out of 800 samples. Overall, total shows 110 misclassification from 4000 samples and 3890 samples correctly identify the model.



True Label \ Predicted Label	Bacteria	Fungi	Healthy	Pests	Virus
Bacteria	784	0	3	10	3
Fungi	4	780	7	9	0
Healthy	2	0	780	17	1
Pests	0	0	27	773	0
Virus	4	0	0	23	773

Fig. 3. Confusion Matrix

Figure 5 shows, the Receiver Operating Characteristic (ROC) curves and corresponding Area Under the Curve (AUC) scores were computed for all five target classes. The ROC curve shows the model ability to distinguish between the classes across varying threshold values. ROC curve shows true and false positive rate. In the ROC curve the top-left corner, indicating near-perfect separability across all classes. In the ROC curve figure also represent the AUC scores where each of the classes AUC score is 1.00. This perfect AUC score signifies that the proposed CNN model consistently achieves high true positive rates while maintaining low false positive rates across all classes.

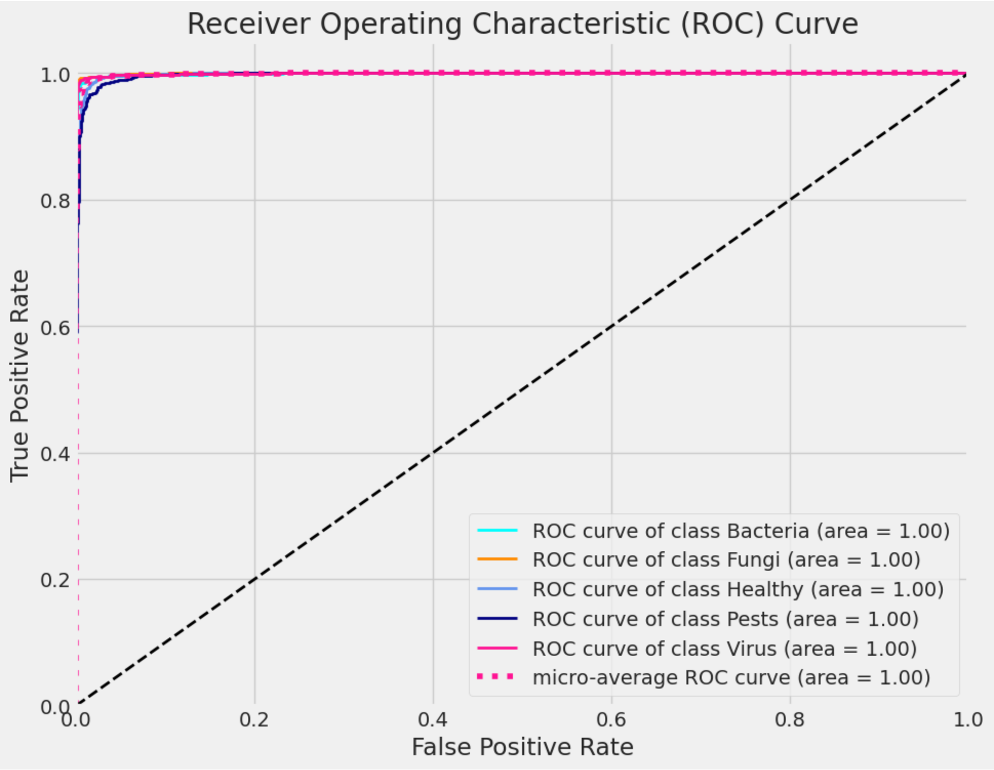


Fig. 4. ROC Curve and AUC Score

The custom CNN model outperformed all three pretrained models in accuracy, achieving 96.88%, compared to 93.92% for VGG16, 86.97% for DenseNet121, and 73.42% for MobileNetV2. While VGG16 showed strong performance, it still slightly behind the custom model. MobileNetV2, designed for lightweight tasks, was the least accurate in this context. Overall, the results show that a well-designed custom CNN can be more effective for specific tasks like plant pathogen classification.

Table II: Comparative Analysis

Model Name	Accuracy Score
VGG16	93.92
MobileNetV2	73.42
DenseNet121	86.97
Custom CNN Model	96.88

Conclusion

In this study, we developed a custom CNN model to help identify plant diseases caused by bacteria, fungi, viruses, pests, and also to recognize healthy plants. The model was designed to work well in real-world situations and showed excellent results, achieving 96.88% test accuracy. It even outperformed well-known pretrained models like VGG16, DenseNet121, and MobileNetV2.

We used smart techniques like data augmentation, batch normalization, dropout, and L2 regularization to improve the model's ability to generalize and avoid overfitting. Evaluation metrics, including precision, recall, F1-score, and AUC, showed that our model was consistent and highly reliable. Overall, our custom CNN model offers a powerful solution for early and accurate plant disease detection.

References

[1] G. N. Agrios, "Plant pathogens and disease: General introduction," in Encyclopedia of Microbiology (3rd ed.), M. Schaechter, Ed. Academic Press, 2009, pp. 613–646. doi: 10.1016/B978-012373944-5.00344-8.

[2] K. Gupta, "Pathogen Dataset," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/kanishk3813/pathogen-dataset

[3] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Computational Intelligence and Neuroscience, vol. 2016, Art. no. 3289801, Jun. 2016, doi: 10.1155/2016/3289801.

[4] D. Sivabalaselvamani, M. D. Selvakarthi, M. L. Rahunathan, and M. P. Leninpugalhanthi, "Detection and Classification of Plant Pathology with Image Processing Technique," (2019).

[5] A. Negi, K. Kumar, and P. Chauhan, "Deep Neural Network-Based Multi-Class Image Classification for Plant Diseases," in [Book Title], A. Choudhury, A. Biswas, M. Prateek, and A. Chakrabarti, Eds. [Publisher], 2021, ch. 6, doi: 10.1002/9781119769231.ch6.

[6] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," Environmental Monitoring and Assessment, vol. 195, no. 1, p. 120, Jan. 2023, doi: 10.1007/s10661-022-10656-x

[7] A. Sinha and R. S. Shekhawat, "A novel image classification technique for spot and blight diseases in plant leaves," Imaging Sci. J., vol. 68, no. 4, pp. 225-239, 2020, doi: 10.1080/13682199.2020.1865652.

[8] G. Wang, Y. Sun, and J. Wang, "Automatic image-based plant disease severity estimation using deep learning," Comput. Intell. Neurosci., vol. 2017, Art. no. 2917536, Jul. 2017, doi: 10.1155/2017/2917536.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520. [Online]. Available: https://doi.org/10.48550/arXiv.1801.04381