ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/639-647
Annu Hari Priya *et. al.*, / International Journal of Engineering & Science Research

Mulberry Leaf Disease Detection Using CNN-Based Smart Android Application

K Shireesha[1], Annu Hari Priya[2], Neelgar Harshitha[3]
[1]Associate professor, Department of CSE, Bhoj Reddy Engineering College for Women, India
[2,3]B.Tech Students, Department of CSE, Bhoj Reddy Engineering College for Women, India

ABSTRACT

Mulberry leaves serve as the primary food source for Bombyx mori silkworms, crucial for silk thread production. However, mulberry trees are highly susceptible to diseases, spreading rapidly and causing significant losses. Manual disease identification across large farms is arduous and time-consuming.

Leveraging computer vision for early disease detection and classification can mitigate up to 90% of production losses. This study collected leaves from two regions of Bangladesh, categorized as healthy, leaf rust-affected, and leaf spot-affected. With a total of 1091 images, split into training (764), testing (218), and validation (109) sets for 5-fold cross-validation, preprocessing and augmentation yielded 6,000 images, including synthetics. This study compares ResNet50, VGG19, and MobileNetV3Small on a specific task following architecture modifications. Four convolutional layers with different output channels (512, 128, 64, and 32) were added to baseline models. We assessed how these architectural changes affected model correctness, computing efficiency, and convergence rates. Comparing three pretrained convolutional neural networks (CNNs) - MobileNetV3Small, ResNet50, and VGG19 - augmented with four additional layers, the modified MobileNetV3Small excelled in precision, recall, F1-score, and accuracy, achieving notable results of 97.0%, 96.4%, 96.4%, and 96.4%, respectively, across cross-validation folds. An efficient smartphone application employing the proposed model for mulberry leaf disease recognition was developed. Overall, the model outperformed existing State of the Art (SOTA) approaches, showcasing its effectiveness in disease identification. The interpretative Grad-CAM visualization images match sericulture specialists' assessments, validating the model's predictions. These results imply that, this explainable AI (XAI) approach with a modified deep learning architecture can appropriately classify mulberry leaves.

1-INTRODUCTION

The mulberry tree (Morus alba), native to northern China, is widely cultivated for its fruit, leaves, and medicinal properties. Mulberry fruits are rich in vitamins, minerals, and antioxidant compounds, while the leaves are vital for silkworms, supporting silk production and offering essential nutrients like protein and chlorophyll. Cotton, a key textile material, and mulberry trees, especially the white mulberry, are crucial to the silk industry. However, fungal and bacterial diseases, such as leaf spot and rust, diminish the nutritional quality of mulberry leaves. A dataset of mulberry images from Bangladesh, annotated into healthy, leaf spot, and leaf rust categories, was used for disease classification research.

This research focuses on developing a deep learning-based solution for detecting diseases in mulberry leaves, specifically leaf spot and leaf rust, which impact the quality of leaves used in silk production. A Transfer Learning-based

Convolutional Neural Network (CNN) model is employed to classify the diseases. Due to the limited availability of annotated mulberry images, data augmentation techniques are used to generate synthetic images to prevent overfitting and improve model performance. The model is lightweight, making it suitable for deployment in mobile applications.

To enhance model transparency, Gradient-weighted Class Activation Mapping (Grad-CAM) is used, which provides interpretability by highlighting the regions of interest in the images. This approach ensures the model focuses on the relevant parts of the leaf for accurate disease classification. The research aims to assist in real-time disease detection, enabling better mulberry plantation management.

2-LITERATURE REVIEW

The integration of deep learning, particularly Convolutional Neural Networks (CNNs), into plant disease detection has brought about substantial progress in agriculture. These CNN-based models have been successfully applied to detect diseases in a wide range of crops, such as tomatoes, maize, oranges, rice, and apples, with accuracy rates consistently exceeding 90%, often reaching up to 99%. For instance, studies like that of Zhou et al., who applied a restructured residual dense network (RRDN), achieved an impressive 95% accuracy in identifying tomato leaf diseases. Models such as EfficientNetB0 and DenseNet121 have also shown remarkable performance in accurately diagnosing unhealthy corn plants, with accuracies of 98.56% and beyond.

However, the effectiveness of these models often comes with a trade-off: the reliance on large datasets and complex neural network architectures, which necessitate considerable computational resources and long training times. This limits the applicability of such models on low-power devices, which are critical in real-world agricultural settings where mobility and quick decision-making are paramount. For mobile and embedded systems, it becomes essential to reduce the complexity of models by designing lightweight alternatives that require fewer parameters, lower memory usage, and faster processing times. Lightweight models like MobileNet and EfficientNet have been designed with these constraints in mind, offering a more feasible solution for deployment in the field, especially for real-time plant disease detection applications.

Additionally, the current landscape of plant disease detection still lacks widespread integration of explainable AI (XAI) techniques. Models like Grad-CAM, which provide visual insights into the decision-making process of a neural network, are largely underutilized in plant disease diagnostics. By using such methods, not only can we improve the transparency and interpretability of the models, but we can also increase the trust that agricultural experts and farmers place in these AI-driven solutions. The ability to visualize which parts of a plant image contribute to a disease classification can enable more informed decision-making, providing actionable insights that could lead to better-targeted interventions. This opportunity for integrating explainable AI into the detection pipeline presents an important avenue for future research, which can enhance the reliability.

3-MATERIALS AND METHODS

Image Acquisition

The dataset for this research was acquired in collaboration with the Bangladesh Sericulture Development Board and includes high-resolution images of mulberry leaves. The dataset contains images of two diseases: leaf rust and leaf spot, along with healthy leaves. These images were captured at different locations in Bangladesh, including Mirganj, Bagha, and Vodra. In total, 1,091 images were categorized by sericulture experts with over ten years of experience. The dataset consists of 308 healthy images, 342 images with leaf rust, and 114 images with leaf spot. Every image in the dataset is high resolution (4,000 × 6,000 pixels).

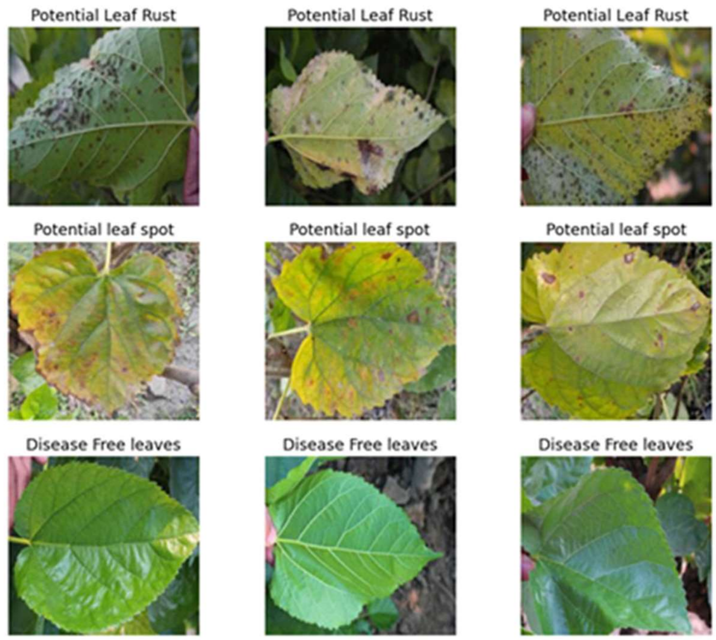


Fig 1 Obtained images**.**

Software Application Development

1. **Android Studio:** Android Studio was used as the integrated development environment (IDE) for developing the Android application. It offers powerful tools for designing the user interface, editing code, and integrating machine learning models, including support for TensorFlow Lite and the Android Neural Networks API.
2. **TensorFlow Lite:** TensorFlow Lite was utilized to enable efficient machine learning on Android devices. It supports model deployment with minimal computational overhead, utilizing hardware acceleration for faster inference. TensorFlow Lite provides tools for model quantization and optimization, ensuring performance without sacrificing accuracy.

4-EXPERIMENTAL RESULTS

Performance Metrics

To evaluate the models, the following performance metrics were used:

- Accuracy (ACC): The ratio of correctly predicted instances to the total instances.
- Precision (P): The proportion of true positive predictions to the total positive predictions made by the model.
- Recall (R): The proportion of true positive predictions to all actual positive instances.
- F1-Score: The harmonic mean of precision and recall, balancing the two metrics.
- Area Under the Curve (AUC): Represents the model's ability to distinguish between classes.

Formulas:

- Accuracy = (TP + TN) / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)

- Recall = TP / (TP + FN)
- F1-Score = 2 × (Recall × Precision) / (Recall + Precision)

Multiclass Classification Results

Model Performance

The study compared the performance of three deep learning models—MobileNetV3Small, VGG19, and ResNet50—for multiclass leaf disease detection. Each model underwent fine-tuning to adapt pre-trained weights to the specific dataset, ensuring optimal performance.

MobileNetV3Small:

- Achieved the highest accuracy of 96.4%, making it the most effective model for disease detection.
- Recorded a precision of 97% and a recall of 96.4%, demonstrating its ability to identify both diseased and healthy leaves accurately.
- Exhibited some variability in performance across validation folds, with a standard deviation of 4.93%, indicating occasional inconsistency.
- The model's lightweight architecture allowed for faster training and inference, making it well-suited for deployment on resource-constrained devices like smartphones.
- After quantization and conversion to TensorFlow Lite (TFLite), it retained 95% of its original accuracy with a reduced computational footprint, ensuring real-world usability.

VGG19:

- Delivered an accuracy of 94.2%, slightly lower than MobileNetV3Small but still competitive.
- Precision and recall were consistent at around 94%, indicating balanced performance across all classes.
- Showed improved stability compared to MobileNetV3Small, with a standard deviation of 3.49% during cross-validation.
- However, VGG19's heavier architecture led to slower training times and higher memory requirements, making it less ideal for mobile deployment.
- Despite its limitations in deployment efficiency, VGG19's feature extraction capabilities made it robust for complex datasets.

5-ANDROID APPLICATION

Android Application

- To demonstrate the practical application of the deep learning model in real-world scenarios, an Android application was developed, integrating the trained CNN model for real-time disease classification of mulberry leaves. The goal was to make the disease detection process more accessible by offering an easy-to-use mobile solution that enables farmers, agricultural experts, and researchers to identify leaf diseases on the go, using just their smartphones. By converting the model into a lightweight, optimized format for mobile devices, the application facilitates rapid and accurate diagnosis without requiring specialized knowledge or equipment.
- **Model Conversion and Optimization for Mobile Devices**
- The model that was initially trained for leaf disease classification, utilizing a convolutional neural network (CNN), was converted into the TensorFlow Lite (TFLite) format to make it compatible with Android devices. TensorFlow Lite is a lightweight version of TensorFlow specifically designed for mobile and embedded devices. It allows models to run efficiently on smartphones by reducing the computational resources required.
- One of the key challenges when deploying deep learning models on mobile devices is the model's size and computational demand. To address this, model quantization techniques were applied, which

reduced the model's size by converting its weights and operations from 32-bit floating-point to 8-bit integer representations. This quantization process not only shrank the model, making it more suitable for mobile devices with limited storage, but also reduced the number of calculations required during inference. As a result, the application offers lower power consumption, faster inference times, and increased efficiency—critical factors when running on devices like smartphones with limited processing capabilities.

- **Workflow and User Interaction**
- The Android application follows a structured workflow for classifying mulberry leaf images. The app initially loads the converted TFLite model into memory and creates a TFLite Interpreter, which acts as an interface between the app and the model. This Interpreter is responsible for feeding the leaf image into the model and generating the classification output. The input image is first resized to 224 x 224 pixels, the standard input size for many deep learning models, and then normalized using the TFLite Image Processor API. This normalization process ensures that the pixel values of the image are in the appropriate range for the model to make accurate predictions.

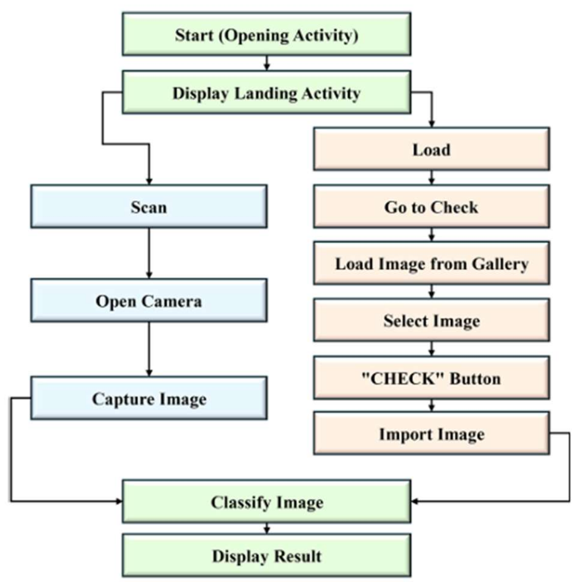


Fig 2 Flowchart of the android application functionality

6-RESULTS

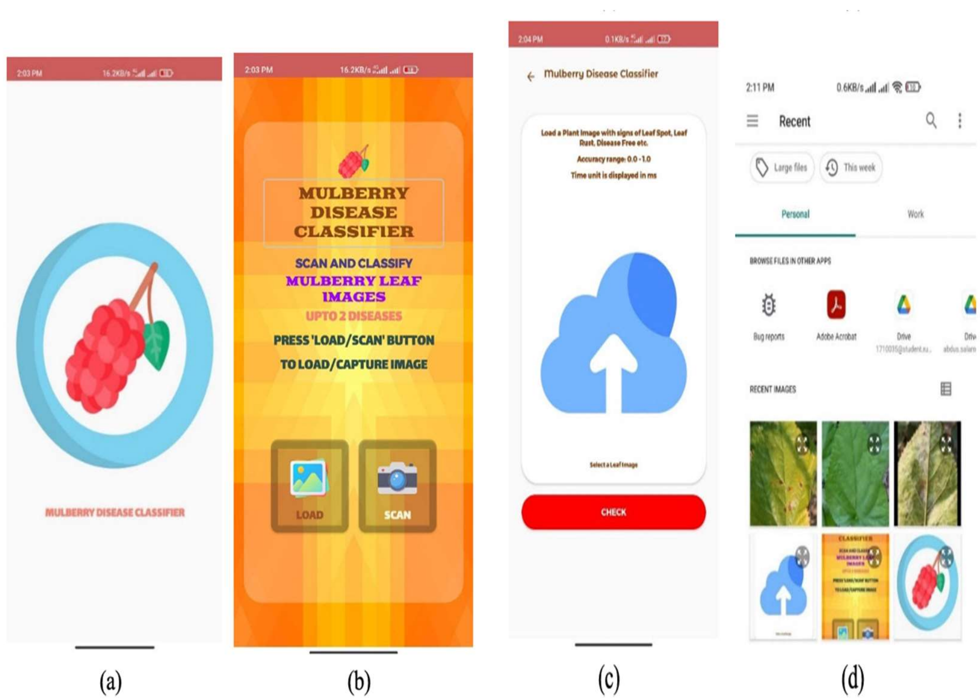


Fig 4.2 Images of the android application activities (a) Opening activity, (b) Landing page, (c) Checking activity, (d) Select image from gallery

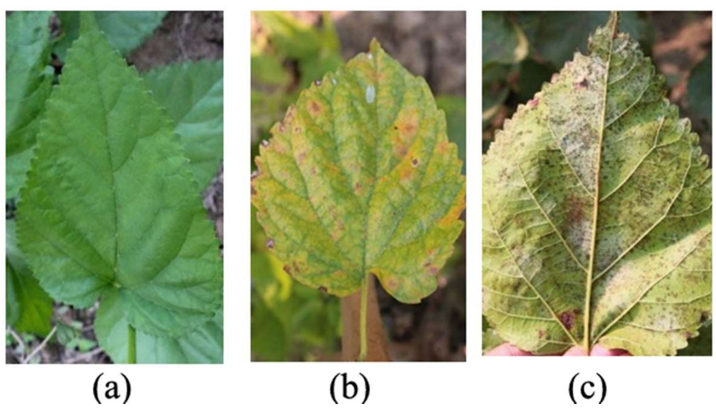
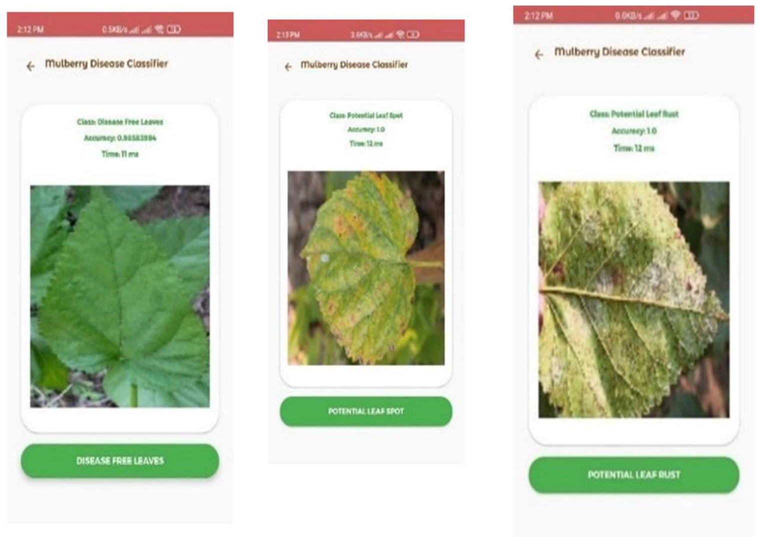


Fig 4.3 Examples of the android App TFLite's classification of images of (a)Disease free leaves, (b)Leaf spot, and (c)Leaf rust input images.



(a) (b) (c)

Fig 4.4 (a), (b), and (c) represents the output from the App for (a)Disease free leaves, (b)Leaf spot, and (c)Leaf ust

Performance and Efficiency

The performance of the Android application was thoroughly tested, with results indicating impressive speed and accuracy. Inference times ranged between 4 and 7 milliseconds for each leaf image classification, showcasing the model's ability to make quick predictions even on mobile hardware. This rapid inference time is achieved by utilizing GPU acceleration during the inference process, although the application can still classify images efficiently without GPU support.

The testing was conducted on a Poco X2 smartphone running Android 11 and equipped with a Qualcomm Snapdragon 730G chipset. The chipset features an Octa-core CPU (2 x 2.2 GHz Kryo 470 Gold & 6 x 1.8 GHz Kryo 470 Silver) and an Adreno 618 GPU, providing a solid foundation for running deep learning models on the device. The tests revealed that the app could classify mulberry leaf images in under 12 milliseconds, which is significantly faster than traditional manual analysis by plant disease specialists.

Energy Efficiency and Power Consumption

An essential consideration in mobile application development is energy efficiency, especially for deep learning applications that can drain a device's battery quickly. By leveraging TensorFlow Lite's optimizations, model quantization, and efficient on-demand inference, the app ensures minimal battery usage while still delivering high performance. The quantization process, in particular, helps reduce the computational load, which in turn reduces the energy consumption during inference.

The application was designed to strike a balance between performance and energy economy, ensuring that users can get accurate predictions without draining their smartphone's battery. This is crucial for ensuring the app is usable in the field, where charging opportunities may be limited.

CONCLUSION AND Future Scope

This study investigated various transfer learning models for mulberry leaf disease detection, including modified versions of MobileNetV3Small, VGG19, and ResNet50. Among these, the modified MobileNetV3Small model exhibited superior accuracy but showed higher performance variability. VGG19 demonstrated robustness, although its accuracy was slightly lower. ResNet50 emerged as a strong contender, offering a well-rounded combination of accuracy and consistency, particularly due to its low standard deviation, making it reliable in diverse conditions.

The choice of the final model for deployment was based on both performance and efficiency, with MobileNetV3Small being selected for its compact size. Its small model size, measured in megabytes, made it ideal for integration into mobile applications. The resulting Android app maintained a size of less than 100MB, a desirable characteristic for mobile apps running machine learning models, ensuring smooth operation without overwhelming device resources.

Although the results indicate the promising potential of the proposed framework for disease detection in mulberry leaves, the study acknowledges the need for further investigation. A key area for future work involves testing the model on additional datasets, such as PlantVillage and PlantDoc, to evaluate its generalizability and adaptability to a wider range of plant diseases. These datasets will provide diverse image quality metrics, disease stages, and disease types, which will challenge the model's ability to handle real-world variability effectively.

Further research should focus on testing the model's robustness across multiple datasets, identifying any potential biases, and enhancing its generalizability to ensure that it remains effective in varied and dynamic agricultural environments. Rigorous and comprehensive testing will pave the way for a more reliable and versatile disease detection system, beneficial for the agricultural sector.

Future Scope

This study emphasizes the importance of artificial intelligence (AI) in addressing agricultural challenges, particularly in the silk production industry, where mulberry trees are susceptible to diseases that can negatively impact productivity. By utilizing transfer learning, the research successfully developed a lightweight disease classification model based on MobileNetV3Small, showing significant progress in detecting diseases affecting mulberry leaves.

The application of Convolutional Neural Networks (CNNs) has proven effective in classifying mulberry leaf diseases, with the integration of explainable AI (XAI) methods like Grad-CAM offering transparency in model decision-making, increasing trust in the system's predictions. The experimental results demonstrate that the proposed model outperforms existing state-of-the-art techniques in plant pathology.

The dataset used in the research is comprehensive, covering both healthy leaves and those affected by diseases like leaf rust and leaf spot, which supports the development of a robust and versatile model. This dataset allows for a thorough investigation and validation of the model across multiple disease categories.

Looking ahead, the automation of mulberry disease classification could benefit from the integration of drone technology. Drones, equipped with high-resolution cameras, can capture real-time aerial images over large crop areas, providing crucial data for early disease detection and continuous monitoring. This would allow for precise tracking of disease progression, ensuring timely intervention and more efficient resource allocation. Such innovations in data collection, alongside the existing mobile applications, can lead to improved silk production output and sustainability. Drones, in particular, could complement static cameras by offering dynamic, large-scale monitoring capabilities, enabling a more comprehensive and efficient approach to crop disease management.

REFERENCES

[1] E. M. Sánchez-Salcedo, P. Mena, C. García-Viguera, J. J. Martínez, and F. Hernández, ''Phytochemical evaluation of white (Morus alba L.) and black (Morus nigra L.) mulberry fruits, a starting point for the assessment of their beneficial properties,'' J. Funct. Foods, vol. 12, pp. 399–408, Jan. 2015.

[2] M. Gundogdu, F. Muradoglu, R. I. G. Sensoy, and H. Yilmaz, ''Determi nation of fruit chemical properties of Morus nigra L., Morus alba L. and Morus rubra L. by HPLC,'' Scientia Horticulturae, vol. 132, pp. 37–41, Dec. 2011.

[3R. V. Kumar, S. Chauhan, D. Kumar, and N. More, ''Nutritional composition in leaves of some mulberry varieties:- a comparative study,'' in Proc. Int. Conf. Bioinf. Biomed. Technol., Apr. 2010, pp. 438–442.

[4] K. Prashar, R. Talwar, and C. Kant, ''CNN based on overlapping pooling method and multi-layered learning with SVM&KNN for American cotton leaf disease recognition,'' in Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM), Apr. 2019, pp. 330–333.