International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 08 | Aug - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

A Mobile-Based Deep Learning Approach for Mango Leaf Disease Detection Using TensorFlow Lite and Flutter Framework.

Mahima Ganapati Bhat[1], Swetha C S[2]

[1]*Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India (1BI23MC071)*
[2]*Assistant Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

---***---

ABSTRACT

Mango cultivation faces significant challenges from leaf diseases such as Gall Midge and Sooty Mould, which can reduce crop yield by up to 40% if not detected early. Traditional disease identification methods rely heavily on expert knowledge and laboratory analysis, which are time-consuming, expensive, and inaccessible to small-scale farmers. This study presents a comprehensive mobile-based solution for automated mango leaf disease detection using deep learning techniques deployed on mobile devices. The system employs a Convolutional Neural Network (CNN) optimized through quantization-aware training and deployed via TensorFlow Lite within a Flutter-based cross-platform mobile application. The dataset comprises 4,962 high-resolution images across three classes: Healthy, Gall Midge, and Sooty Mould, collected from agricultural research stations and field conditions.. The Flutter application enables farmers to capture leaf images through their smartphone cameras and receive immediate disease classification results with confidence scores, detailed disease information, and treatment recommendations. The system operates entirely offline, making it accessible in remote agricultural areas with limited internet connectivity. Results demonstrate that mobile-based deep learning can provide accurate, fast, and accessible plant disease detection, contributing to early intervention, reduced crop losses, and improved agricultural productivity for mango farmers globally.

Keywords: Mango Leaf Disease Detection, TensorFlow Lite, Flutter Framework, Mobile Deep Learning, CNN Quantization, Agricultural Technology, Plant Disease Classification, On-Device Inference

I. INTRODUCTION

Mango (Mangifera indica) is one of the world's most economically important fruit crops, with global production exceeding 50 million tons annually. However, mango cultivation faces persistent challenges from various leaf diseases that significantly impact yield and fruit quality. Among these, Gall Midge (Procontarinia pustulata) and Sooty Mould infections represent two of the most destructive diseases affecting mango orchards worldwide.

Gall Midge disease causes characteristic raised galls or blisters on leaf surfaces, with larvae feeding within plant tissue, leading to leaf deformation, reduced photosynthesis, and premature leaf drop. In severe infestations, over 100 galls can develop on a single leaf, potentially causing yield losses ranging from 30-50% in affected regions. Sooty Mould, caused by various fungal species that grow on honeydew secreted by sap-feeding insects, creates dark grey to black coatings on leaf surfaces, interfering with photosynthesis and overall plant health.

Traditional disease identification methods involve manual inspection by agricultural experts, laboratory analysis, or consultation with extension services. These approaches, while accurate, are time-consuming, expensive, and often inaccessible to smallholder farmers in remote areas. The subjective nature of visual assessment can lead to inconsistent diagnoses, while laboratory testing may take several days, potentially allowing diseases to spread and cause irreversible damage.

This research addresses these challenges by developing a comprehensive mobile-based solution for mango leaf disease detection. The system combines state-of-the-art deep learning techniques with mobile optimization.

II. LITERATURESURVEY

The application of deep learning techniques for plant disease detection has gained significant momentum in recent years, with researchers exploring various approaches to address the challenges of automated agricultural diagnosis.

Traditional Machine Learning Approaches:

Early work in plant disease detection primarily relied on traditional machine learning techniques combined with handcrafted feature extraction methods. Studies by Yaligar and Gowda demonstrated the potential of Support Vector Machines (SVM) and Random Forest classifiers for disease classification, though these approaches required extensive domain expertise for feature engineering and showed limited generalization across different imaging conditions.

Mobile and Edge Deployment:

The transition from cloud-based to mobile-deployed models has been a critical focus area. Laxamana et al. developed a TensorFlow Lite-based sugarcane disease classification system achieving 95.6% accuracy while enabling real-time mobile inference. The work by Kumar et al. explored resource optimization techniques, demonstrating that even modest reductions in RAM usage (0.5%) and flash memory (1.14%) could result in significant accuracy improvements (9%) for mobile deployment.

Cross-Platform Mobile Development:

Flutter framework has emerged as a popular choice for agricultural mobile applications due to its cross-platform capabilities and native performance. Research by Patel et al. and Singh et al. demonstrated successful integration of TensorFlow Lite models within Flutter applications, achieving near real-time inference with minimal latency across both Android and iOS platforms.

Gaps in Current Research:
Despite significant progress, several gaps remain in the literature. Most studies focus on single-crop applications with limited real-world deployment validation. Additionally, few researchers have addressed the specific challenges of developing user-friendly interfaces for diverse agricultural communities or provided comprehensive offline functionality for resource-constrained environments.

This research builds upon the established foundation while addressing these gaps through the development of a practical, deployable mobile system specifically designed for mango leaf disease detection with comprehensive offline capabilities and farmer-centric user interface design.

III. PROPOSED METHEDOLOGY

This research addresses these challenges by developing a comprehensive mobile-based solution for mango leaf disease detection. The system combines state-of-the-art deep learning techniques with mobile optimization strategies, deployed through a user-friendly Flutter application that operates entirely offline. The primary contributions of this work include: (1) development of an optimized CNN architecture for three-class mango disease classification, (2) implementation of quantization-aware training for mobile deployment, (3) creation of a cross-platform Flutter application with intuitive user interface, and (4) comprehensive evaluation under real-world conditions with practical deployment considerations.

3.1 Data Collection and Preaparation

Dataset Acquisition:
The dataset was compiled from multiple sources to ensure diversity and real-world applicability. Images were collected from agricultural research stations, commercial mango orchards, and field surveys across different geographic regions. The dataset comprises 4,962 high-resolution images (minimum 1024×768 pixels) representing three distinct classes: Healthy leaves, Gall Midge infection, and Sooty Mould infestation.

Data Distribution and Characteristics:
The collected dataset maintains relatively balanced class distribution to prevent bias during training. Healthy leaf images (1,715 samples) represent various mango varieties under different lighting conditions and growth stages. Gall Midge samples (1,592 images) capture different severity levels of infestation, from early-stage small galls to severe cases with multiple lesions per leaf. Sooty Mould images (1,655 samples) document various stages of fungal growth, from light surface coverage to complete leaf blackening.

3.2 Data Preprocessing

Quality Assessment: All images underwent manual inspection to ensure proper labeling and remove corrupted or ambiguous samples.

Standardization: Images were resized to 224×224 pixels to match CNN input requirements while maintaining aspect ratios through center cropping or padding.

Normalization: Pixel values were normalized to range by

dividing by 255, ensuring consistent input scaling across different camera sensors and lighting conditions.

Data Augmentation: Training images underwent various augmentation techniques including rotation (±15°), horizontal flipping, brightness adjustment (±20%), and slight zoom variations (0.8-1.2x) to improve model generalization and reduce overfitting.

Train-Validation-Test Split: The dataset was partitioned into training (72.7%), validation (18.2%), and test (9.1%) sets, maintaining class balance across all splits.

3.3 CNN Architecture Design

Base Architecture Selection:
After comprehensive evaluation of multiple CNN architectures including ResNet50, MobileNetV2, and EfficientNet-B0, a modified MobileNetV2 architecture was selected for optimal balance between accuracy and mobile deployment requirements.
Model Architecture Components:

Input Layer: Accepts 224×224×3 RGB images with batch normalization for stable training.

Feature Extraction Block: Modified MobileNetV2 backbone with pre-trained ImageNet weights, fine-tuned for agricultural imagery. The architecture includes inverted residual blocks with linear bottlenecks optimized for mobile deployment.

Classification Head: Custom classification layers comprising global average pooling, dropout (0.3), dense layer (128 units), and final softmax layer for three-class classification.

Regularization: Batch normalization, dropout layers, and L2 regularization (1e-4) were implemented to prevent overfitting and improve generalization.

3.4 Model Training

Training Configuration:

The model was trained using adaptive learning rate scheduling with initial learning rate of 1e-3, decaying by factor of 0.1 when validation loss plateaued. Training employed categorical cross-entropy loss with Adam optimizer, utilizing early stopping based on validation accuracy with patience of 10 epochs.

Quantization-Aware Training:

To optimize for mobile deployment, quantization-aware training was implemented using TensorFlow Model Optimization toolkit. This approach simulates quantization effects during training, allowing the model to adapt to reduced precision while maintaining accuracy.

Performance Evaluation:

Model performance was evaluated using standard classification metrics including accuracy, precision, recall, and F1-score, calculated individually for each class and aggregated using

macro-averaging. The confusion matrix analysis provided detailed insights into classification patterns and potential misclassification trends.

3.5 Mobile Optimization and TensorFlow Lite Conversion

Model Conversion Pipeline: The trained model underwent optimization for mobile deployment through TensorFlow Lite conversion with post-training quantization. The process included:

Graph Optimization: Unnecessary operations were removed, and mathematical operations were fused for improved inference efficiency.

Weight Quantization: Model weights were quantized from 32-bit floating-point to 8-bit integers, significantly reducing model size while maintaining acceptable accuracy.

Activation Quantization: Input and output tensors were quantized to enable full integer inference on mobile devices

3.6 Flutter Application Development

Cross-Platform Framework Selection: Flutter was chosen for application development due to its single codebase deployment across Android and iOS platforms, native performance characteristics, and rich ecosystem of machine learning integration packages.

Application Architecture:

The Flutter application follows a layered architecture pattern:

Presentation Layer: User interface components including home screen, camera interface, and results display implemented using Flutter's widget system.

Business Logic Layer: Image processing, model inference coordination, and result interpretation logic implemented using provider pattern for state management.

Data Layer: TensorFlow Lite model integration, local storage management, and offline resource access through platform-specific implementations.

Key Features Implementation:

Camera Integration: Seamless camera access for image capture with preview functionality and image quality validation.

Real-time Inference: On-device model inference with progress indicators and error handling for various edge cases.

Results Visualization: Confidence score display through interactive charts, detailed disease information, and treatment recommendations.

Offline Functionality: Complete offline operation including model inference, disease information access, and result storage.

IV. Results and Analysis

4.1 Model Performance Evaluation

The developed CNN architecture demonstrated strong performance across all evaluation metrics, achieving competitive results suitable for practical agricultural applications.

Overall Classification Performance:

The quantized model achieved an overall accuracy of 93.8% on the test dataset, with precision of 93.2%, recall of 94.5%, and F1-score of 93.8%. These results indicate robust classification capability while maintaining the computational efficiency required for mobile deployment.

Class-Specific Performance Analysis:

Individual class performance revealed consistent accuracy across all three categories:

Healthy Class: Achieved 96.2% precision and 95.9% recall, indicating excellent ability to correctly identify disease-free leaves while minimizing false positive diagnoses.

Gall Midge Class: Demonstrated 91.7% precision and 93.8% recall, showing reliable detection of characteristic gall formations despite potential variations in symptom presentation.

Sooty Mould Class: Attained 92.1% precision and 93.9% recall, effectively distinguishing the distinctive dark mold patterns from other leaf abnormalities.

4.2 Comparative Analysis with Existing Approaches

Architecture Comparison:
Comparative evaluation against standard CNN architectures demonstrated the effectiveness of the optimized approach. While EfficientNet-B0 achieved slightly higher accuracy (95.7%), the proposed quantized model provided superior mobile deployment characteristics with only 1.9% accuracy trade-off for significant improvements in inference speed and model size.

Performance vs. Efficiency Trade-offs:

The quantization-aware training approach successfully balanced accuracy preservation with mobile optimization requirements. Compared to the original full-precision model, the quantized version maintained 98.9% of the original accuracy while providing substantial improvements in deployment metrics.

4.3 Mobile Application Performance

Inference Speed Analysis:

On-device inference performance was evaluated across different Android device specifications:

High-end devices (8GB RAM, Snapdragon 855+): Average inference time 180 ms

Mid-range devices (4GB RAM, Snapdragon 660): Average inference time 245 ms

Budget devices (3GB RAM, MediaTek Helio P60): Average inference time 320 ms

All devices maintained acceptable real-time performance for practical field applications.

Memory Usage and Storage:

The application demonstrated efficient resource utilization with total APK size of 28.7 MB, including the TensorFlow Lite model (8.9 MB), Flutter framework components, and application assets. Runtime memory usage peaked at 145 MB during inference operations, well within acceptable limits for target device specifications.

User Experience Metrics:

Beta testing with 30 agricultural professionals and farmers revealed high user satisfaction scores:

Interface Usability: 4.6/5.0 average rating

Result Accuracy Perception: 4.4/5.0 average rating

Overall Application Utility: 4.5/5.0 average rating

4.4 Real-World Deployment Validation

Field Testing Results:

Extensive field testing across different mango orchards validated the system's practical applicability. The application successfully operated under various environmental conditions, including different lighting scenarios, leaf orientations, and background complexities.

Offline Functionality Assessment:

The complete offline operation capability was verified through testing in areas with limited or no internet connectivity. All core functionalities, including disease classification, confidence scoring, and information display, operated successfully without network dependencies.

Cross-Platform Consistency:

Testing across Android and iOS platforms confirmed consistent performance and user experience, validating Flutter's cross-platform deployment advantages for agricultural applications.

V. FUTURE WORK

Several promising directions emerge from this research that could enhance the system's capabilities and expand its practical impact in agricultural settings.

Dataset Expansion and Diversification:

Future work should focus on expanding the dataset to include additional mango leaf diseases such as Anthracnose, Bacterial Canker, and Powdery Mildew. Incorporating diverse geographical regions, mango varieties, and seasonal variations would improve model generalization and practical applicability across different agricultural contexts.

Multi-Modal Data Integration:

Integration of complementary data sources could significantly enhance diagnostic accuracy. Incorporating environmental sensor data (temperature, humidity), soil conditions, and temporal disease progression patterns could provide contextual information for more comprehensive disease assessment and treatment recommendations.

Advanced Model Architectures:

Exploration of newer CNN architectures such as Vision Transformers (ViTs) and EfficientNet variants specifically optimized for mobile deployment could potentially improve accuracy while maintaining computational efficiency. Additionally, implementing ensemble methods combining multiple specialized models could enhance robustness across different disease presentations.

Real-Time Disease Progression Monitoring:

Development of longitudinal tracking capabilities would enable farmers to monitor disease progression over time, assess treatment effectiveness, and optimize intervention strategies. This would require implementing sophisticated image registration and temporal analysis algorithms.

Edge Computing Integration:

Expanding beyond individual mobile devices to edge computing infrastructures could enable more sophisticated analysis while maintaining local processing benefits. This approach could support collaborative diagnosis across farming communities and enable knowledge sharing without compromising data privacy.

Predictive Analytics and Early Warning Systems:

Integration of predictive modeling capabilities could forecast disease outbreaks based on environmental conditions, historical patterns, and early symptom detection. Such systems could enable proactive management strategies, potentially preventing disease spread before visible symptoms appear.

Multilingual and Multimodal Interface Support:

Enhancing the application with comprehensive multilingual support and voice-based interaction capabilities would improve accessibility for diverse farming communities. Integration of text-to-speech and speech-to-text functionalities could particularly benefit users with limited literacy levels.

Integration with Agricultural Management Systems:

Developing APIs and integration capabilities with existing farm management software, government agricultural databases, and supply chain systems could create comprehensive agricultural health monitoring ecosystems.

VI. CONCLUSION

This research successfully demonstrates the feasibility and effectiveness of mobile-based deep learning for automated mango leaf disease detection. The developed system addresses critical challenges in agricultural disease diagnosis by providing accurate, accessible, and real-time diagnostic capabilities directly to farmers and agricultural professionals.

The quantized CNN model achieved strong performance metrics with 93.8% accuracy, 93.2% precision, 94.5% recall, and 93.8% F1-score, validating the effectiveness of the deep learning approach for mango leaf disease classification. The successful implementation of quantization-aware training resulted in an 80.3% reduction in model size while maintaining over 98% of the original accuracy, demonstrating the viability of complex AI models for resource-constrained mobile deployment.

The Flutter-based cross-platform application successfully provides comprehensive offline functionality, enabling deployment in remote agricultural areas with limited internet connectivity. The intuitive user interface design, validated through field testing with agricultural professionals, confirms the system's practical applicability for diverse user populations.

Key technical contributions include the development of an optimized CNN architecture specifically tailored for mango disease classification, implementation of effective model quantization strategies for mobile deployment, and creation of a robust cross-platform mobile application with comprehensive offline capabilities. The system's ability to operate entirely on-device while providing immediate diagnostic feedback represents a significant advancement in accessible agricultural technology.

The research addresses real-world agricultural challenges by democratizing access to expert-level diagnostic capabilities, potentially reducing crop losses, and improving decision-making for mango farmers globally. The offline-first design ensures accessibility in resource-constrained environments, while the cross-platform deployment maximizes reach across different mobile ecosystems.

Future enhancements focusing on dataset expansion, multi-modal data integration, and predictive analytics capabilities will further strengthen the system's impact on sustainable agricultural practices. The established foundation provides a scalable platform for extending diagnostic capabilities to additional crops and diseases, contributing to broader agricultural technology advancement.

This work represents a meaningful step toward leveraging artificial intelligence for practical agricultural applications, demonstrating that sophisticated deep learning capabilities can be successfully deployed in resource-constrained environments to address critical real-world challenges. The combination of technical innovation with practical deployment considerations provides a model for future agricultural AI applications, contributing to global food security and sustainable farming practices.

VII. REFERNCES

[1]. Gulavnai, Sampada, and Rajashri Patil. "Deep learning for image based mango leaf disease detection." International Journal of Recent Technology and Engineering 8.3S3 (2019): 54-56.

[2]. Sharma, Amisha, et al. "Mango leaf diseases detection using deep learning." International Journal of Knowledge Based Computer Systems 10.1 (2022): 40-44.

[3]. Pratap, V. Krishna, and N. Suresh Kumar. "Deep Learning based Mango Leaf Disease Detection for Classifying and Evaluating Mango Leaf Diseases." Fusion: Practice & Applications 15.2 (2024).

[4]. Rajpoot, Vikram, et al. "Mango plant disease detection system using hybrid BBHE and CNN approach." Traitement du Signal 39.3 (2022): 1071-1078.

[5]. Gining, R. A. J. M., et al. "Harumanis mango leaf disease recognition system using image processing technique." Indonesian Journal of Electrical Engineering and Computer Science 23.1 (2021): 378-386.

[6]. Thaseentaj, Shaik, and S. Sudhakar Ilango. "Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations." International Journal of Intelligent Engineering & Systems 17.4 (2024).

[7]. Sandhya, S., A. Balasundaram, and S. Arunkumar. "Deep learning and computer vision based model for detection of diseased mango leaves." International Journal on Recent and Innovation Trends in Computing and Communication 10.6 (2022): 70-79.

[8]. Jain, Sachin, and Preeti Jaidka. "Mango leaf disease classification using deep learning hybrid model." 2023 International Conference on Power, Instrumentation, Energy and Control (PIECON). IEEE, 2023.

[9]. Sharma, Rohan, et al. "Detecting diseases in mango leaves using convolutional neural networks." International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 1. Singapore: Springer Singapore, 2021.

[10]. Patel, Aastha, et al. "Utilizing TFLite and machine learning for the early detection of mango leaf disease: an automated flutter application." 2024 5th International Conference for Emerging Technology (INCET). IEEE, 2024.