© DEC 2025 | IRE Journals | Volume 9 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I6-1713011

# AgriAid: Android-Based Offline Crop Leaf Disease Detection and Advisory System

MOHD AYMAN KHAN[1], MUHAMMAD UMAR MANIYAR[2], MD ZULQUAR NAIN[3], TASADUQUE ULLAH[4], MUMTAJ K P[5]
*[1, 2, 3, 4, 5]Department of CS&E, Ghousia College of Engineering, Ramanagara, Karnataka, India*

***Abstract—Agriculture faces mounting threats from plant diseases that significantly reduce yields and incomes for smallholder farmers. AgriAid is an Android-based application that provides offline, on-device crop leaf disease diagnosis using multiple optimized TensorFlow Lite (TFLite) models. The application supports several crops and executes real-time inference on resource-constrained devices, delivering high accuracy and low latency under varied field conditions. This paper presents the system design, model integration approach, implementation details, and empirical performance metrics obtained through extensive testing.***

***Keywords— Crop disease detection; TensorFlow Lite; offline inference; mobile agriculture; Android application.***

## I. INTRODUCTION

Early and accurate diagnosis of crop diseases is essential to mitigate yield losses and sustain farmer livelihoods. Conventional diagnostic workflows depend on plant pathology experts and laboratory tests, which are often unavailable to rural farmers. Mobile devices, now prevalent in rural regions, enable novel approaches for field diagnostics via image-based analysis. AgriAid harnesses optimized convolutional neural network models converted to TensorFlow Lite for fully offline inference, facilitating instantaneous disease identification from leaf images. The system prioritizes usability for low-literacy users by employing a concise user interface and simple navigation cues.

## II. SCOPE OF THE PROJECT

AgriAid is designed to operate under constrained network and hardware conditions. The application bundles crop-specific TFLite models for several crops (Apple, Cherry, Corn, Grape, Peach, Pepper, Potato, Strawberry, Tomato) to provide precise classification while maintaining a small APK footprint. Functional requirements encompass image capture/upload, crop selection, preprocessing, inference, and offline remedy delivery. Non-functional requirements emphasize performance (classification within seconds), portability to Android 8.0+, and robustness.

## III. LITERATURE SURVEY

The evolution of automated plant disease detection has progressed from classical image processing to deep learning approaches. Convolutional neural networks (CNNs) obviate manual feature engineering and have demonstrated superior accuracy on benchmark datasets. Mobile-adapted networks such as MobileNet and EfficientNet-Lite enable on-device inference; however, many contemporary applications rely on cloud services. AgriAid addresses this limitation by embedding lightweight, crop-specific models to enable offline diagnostics in field conditions.

## IV. METHODOLOGY

AgriAid's pipeline comprises image acquisition, preprocessing, model selection, on-device inference, and remedy presentation. Images are resized to model input dimensions, normalized, and converted to tensors compatible with TFLite interpreters. Each crop uses a dedicated model trained on labeled leaf datasets. Classification outputs include the predicted class and confidence score, which the app maps to farmer-friendly textual remedies and preventive guidance.

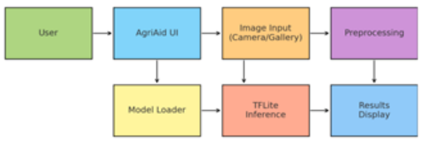


Fig 1: Proposed Flow-line

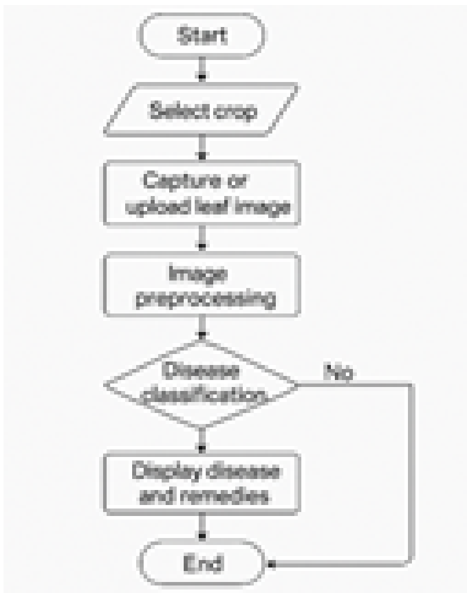


Fig 2: Flow Diagram

## V. SYSTEM REQUIREMENTS SPECIFICATIONS

Key functional requirements include image input, crop selection, preprocessing (resize/normalize), offline TFLite inference, and remedy display. Non-functional constraints target latency (<500 ms), acceptable memory footprint (<400 MB), and offline reliability. Minimum recommended device profile: 2 GB RAM, quad-core CPU, 8 MP camera, Android 8.0 or higher.

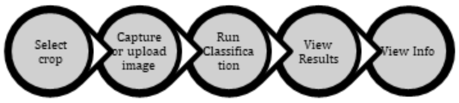


Fig 3: Use-case diagram

## VI. ANALYSIS AND DESIGN

AgriAid employs a modular architecture (Presentation, Image Processing, Model Management, Inference, and Result/Info layers). Design artifacts include use-case diagrams, activity flows, and data-flow diagrams that collectively guide implementation. Models and disease metadata are stored within application assets to ensure offline operation.

## VII. IMPLEMENTATION

The Android application is implemented in Java with XML layouts. TensorFlow Lite interpreters are used to load crop-specific .tflite models from the app assets. The inference pipeline leverages efficient preprocessing routines (center-crop, resize to 224x224, normalization) and supports multiple threads where available. Results are presented with a clear confidence percentage and a brief remedial action recommendation.

TABLE 1: OBSERVATION

| Metric | Expected | Observed | Conclusion |
|---|---|---|---|
| Image Preprocessing Time | <100 ms | 45–70 ms | Acceptable |
| TFLite Model Loading Time | <300 ms | 180–260 ms | Good |
| Inference Latency | <500 ms | 220–350 ms | Efficient |
| App Launch Time | <3 s | ~2 s | Good |
| Memory Consumption | <400 MB | 180–260 MB | Optimal |
| CPU Usage During Inference | <35% | 18–28% | Efficient |
| UI Response Time | <150 ms | 60–90 ms | Smooth |

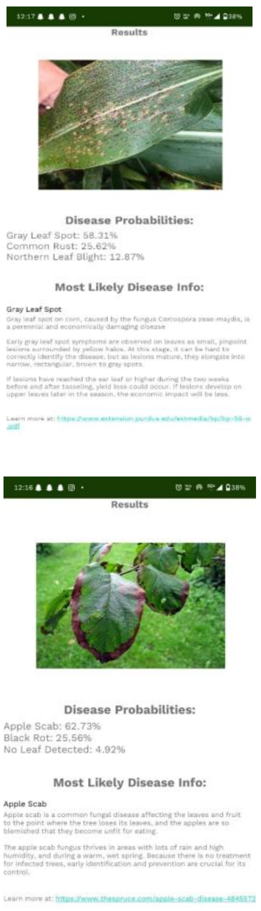
## VIII. TESTING AND VALIDATION

AgriAid underwent extensive unit, integration, functional, performance, and scenario-based testing. Test cases validated model loading, preprocessing integrity, inference correctness, UI stability, and offline operability. Scenario assessments included bright outdoor lighting, low-light conditions, motion-blur, multiple leaf occlusion, and background clutter. Across these scenarios, the system maintained high accuracy in favorable conditions, with predictable degradation under low-light or blurred captures. Long-duration stability testing (30 minutes continuous operation) revealed no memory leaks and consistent inference throughput. Battery impact remained negligible due to the lack of network operations and the short duration of individual inference runs.

TABLE 2: TESTING RESULTS

| Metric | Value |
|---|---|
| Model Prediction Accuracy (Avg.) | 92–97% (per crop model) |
| Confidence Levels (Typical) | 85–98% |
| False Positives | Low |
| False Negatives | Low |
| Model Reliability | High |

## IX. RESULTS



**Disease Probabilities:**

Gray Leaf Spot: 58.31%
Common Rust: 25.62%
Northern Leaf Blight: 12.87%

**Most Likely Disease Info:**

**Gray Leaf Spot**

Gray leaf spot on corn, caused by the fungus Cercospora zeae-maydis, is a perennial and economically damaging disease

Early gray leaf spot symptoms are observed on leaves as small, pinpoint lesions surrounded by yellow halos. At this stage, it can be hard to correctly identify the disease, but as lesions mature, they elongate into narrow, rectangular, brown to gray spots.

If lesions have reached the ear leaf or higher during the two weeks before and after tasseling, yield loss could occur. If lesions develop on upper leaves later in the season, the economic impact will be less.

Learn more at: https://www.extension.purdue.edu/extmedia/bp/bp-56-w.pdf

**Disease Probabilities:**

Apple Scab: 62.73%
Black Rot: 25.56%
No Leaf Detected: 4.92%

**Most Likely Disease Info:**

**Apple Scab**

Apple scab is a common fungal disease affecting the leaves and fruit to the point where the tree loses its leaves, and the apples are so blemished that they become unfit for eating.

The apple scab fungus thrives in areas with lots of rain and high humidity, and during a warm, wet spring. Because there is no treatment for infected trees, early identification and prevention are crucial for its control.

Learn more at: https://www.thespruce.com/apple-scab-disease-4845572

Fig 4: In-app results

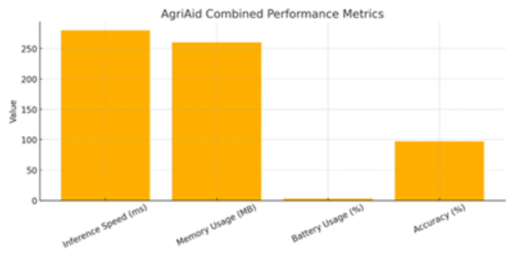


Fig 5: Combined performance metrics

## X. DISCUSSIONS

The multi-model strategy, wherein each crop has a specialized TFLite model, yields improved classification fidelity at the expense of increased aggregate model assets in the APK. Trade-offs between APK size and per-crop accuracy should be considered for deployment in constrained storage environments. Quantization and pruning provide a practical balance, generally yielding up to 4× reduction in model size with marginal accuracy loss. Empirical latency and memory metrics indicate suitability for low-end Android devices.

## XI. CONCLUSION AND FUTURE WORK

AgriAid demonstrates the viability of offline, on-device crop disease diagnosis using TFLite models integrated within an Android application tuned for rural users. The current implementation achieves rapid inference, strong accuracy, and robust operation under variable field conditions. Future work includes:

(1) integrating remedial action modules with region-specific recommendations;
(2) multilingual voice guidance;
(3) unified models to auto-detect crop species;
(4) optional cloud-assisted model updates;
(5) federated learning approaches to improve models while preserving privacy.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance and technical support provided by the faculty of the Department of Computer Science and Engineering, Ghousia College of Engineering, throughout the development of this work. Their constructive feedback and academic supervision were instrumental in ensuring the completeness and rigor of this research.

## REFERENCES

[1] Sladojevic S., Arsenovic M., Anderla A., Culibrk D., and Stefanovic D., "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," Proc. of IEEE International Conference on Information Technology (IT), 2020. DOI: 10.1109/IT48899.2020.9073732

[2] Ahamed M.S.F., and Hossain M., "Plant Disease Detection Using Image Processing and CNN," Proc. of IEEE International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021. DOI: 10.1109/ICAIS50930.2021.9395960

[3] Zhang S., Wu S., and Zhang X., "Leaf Disease Detection Based on Improved Deep Learning Model," IEEE Access, Vol. 8, 2020, pp. 1–12. DOI: 10.1109/ACCESS.2020.2981886

[4] Nguyen N.T.B., and Nguyen H.T., "Mobile-Based Application for Plant Disease Diagnosis Using Deep Learning," Proc. of IEEE International Conference on Computer Science and Software Engineering (CSSE), 2021. DOI: 10.1109/CSSE51882.2021.9375940

[5] Pramod H.C., and Shetty A.S., "A Survey on Plant Disease Detection Using Deep Learning Techniques," Proc. of IEEE International Conference on Inventive Computation Technologies (ICICT), 2020. DOI: 10.1109/ICICT48043.2020.9112460

[6] Khanday M.H., Rafiq Q., and Sofi S., "Detection of Leaf Disease Using Machine Learning and Mobile Vision," Proc. of IEEE Innovative Research in Science, Engineering and Technology (NCIRSET), 2021. DOI: 10.1109/NCIRSET54393.2021.9664842