

CaneCare: Real-Time Sugarcane Leaf Disease Detection Using Deep Learning — EfficientNet-B0, ResNet-50, and MobileNetV3 with Mobile Deployment

Roobannidhi K A , Revin Kumar R, Sakthi GaneshB

Dept. of Computer Science and Engineering RVS Technical Campus, Coimbatore

Abstract- Agriculture plays a vital role in the Indian economy, with sugarcane being one of the major commercial crops. Sugarcane is affected by diseases such as Mosaic, Red Rot, Rust, and Yellow Leaf Disease, which significantly reduce crop yield and quality. Early detection is essential for effective crop management, yet traditional manual inspection methods are time-consuming and inaccessible to rural farmers who lack expert guidance. This paper presents CaneCare, a deep learning-based system for automatic detection and classification of sugarcane leaf diseases from images. A curated dataset of 2,521 images across five classes—Healthy, Mosaic, RedRot, Rust, and Yellow—was used to train and evaluate three convolutional neural network architectures: EfficientNet-B0, ResNet-50, and MobileNetV3, all fine-tuned using transfer learning. ResNet-50 achieved the highest test accuracy of 94.71% (F1 = 0.947), while MobileNetV3 achieved 94.18% accuracy with only 1.5M parameters, making it the preferred choice for mobile deployment. EfficientNet-B0 provided balanced performance at 90.34%. A complete end-to-end deployment pipeline was implemented comprising a FastAPI backend hosted on the Render cloud platform and a React Native mobile application (CaneCare) built with Expo. Grad-CAM visualisations confirm that model predictions are grounded in biologically meaningful, disease-specific leaf features. The system provides real-time disease predictions with confidence scores and treatment recommendations, demonstrating a practical, accessible, and scalable pathway for AI-driven precision agriculture in resource-constrained environments.

Index-Terms- Sugarcane disease detection; convolutional neural networks; transfer learning; EfficientNet-B0; ResNet-50; MobileNetV3; Grad-CAM; mobile application; FastAPI; precision agriculture.

I. INTRODUCTION

Sugarcane (*Saccharum officinarum*) is one of the most economically significant cash crops in India, accounting for a substantial share of agricultural output. Despite its importance, sugarcane cultivation is persistently threatened by plant diseases including Mosaic, Red Rot (*Colletotrichum falcatum*), Rust (*Puccinia melanocephala*), and Yellow Leaf Disease (Sugarcane yellow leaf virus). Unchecked, these diseases cause severe yield losses and significant economic damage for farming communities [1].

Traditional diagnosis depends on visual inspection by experienced agricultural experts—an approach that is subjective, slow, and largely unavailable in rural farming environments. Delayed or incorrect diagnosis leads to inappropriate treatment, excessive pesticide application, and reduced productivity. Accessible, automated diagnostic tools are therefore a pressing need in modern agriculture [2].

Recent advances in deep learning, particularly convolutional neural networks (CNNs), have demonstrated remarkable performance in plant

disease classification from images [3][4]. Transfer learning enables high accuracy even from relatively small domain-specific datasets by leveraging representations pre-trained on large-scale image corpora. Mobile-optimised architectures such as MobileNetV3 further extend these capabilities to resource-constrained devices, enabling deployment directly in the hands of farmers via smartphones.

This paper makes the following contributions:

1. A comprehensive comparative evaluation of EfficientNet-B0, ResNet-50, and MobileNetV3 for sugarcane leaf disease classification using a curated 2,521-image dataset under consistent training conditions.
2. A two-phase transfer learning strategy that achieves 94.71% test accuracy (ResNet-50) and 94.18% (MobileNetV3) with only 1.5M parameters—establishing MobileNetV3 as the optimal mobile deployment model.
3. A complete end-to-end system integrating a FastAPI prediction backend with a React Native mobile application (CaneCare) providing real-time predictions with treatment recommendations.
4. Grad-CAM interpretability analysis confirming that model predictions are grounded in disease-specific leaf features rather than spurious background correlations.

The remainder of this paper is organised as follows. Section II surveys related work. Section III describes the dataset and methodology. Section IV details system implementation. Section V presents experimental results and discussion. Section VI concludes with future directions.

II. RELATED WORK

A. Conventional Machine Learning Approaches

Before the dominance of deep learning, plant disease classification relied on handcrafted feature extraction combined with classical classifiers including Support Vector Machines (SVMs), k-Nearest Neighbours (k-NN), and Random Forest. Feature engineering methods such as colour histograms and Gray-Level Co-occurrence Matrix (GLCM) texture descriptors provided moderate

success under controlled conditions. However, these methods lacked robustness to variations in lighting, background, and image scale, severely limiting their real-world applicability [5].

B. Deep Learning-Based Approaches

Deep CNNs transformed plant disease detection by enabling automatic hierarchical feature learning directly from raw images. Architectures including AlexNet, VGGNet, and ResNet [6] progressively improved classification accuracy on benchmark datasets. EfficientNet [7] introduced compound scaling to simultaneously optimise network depth, width, and resolution. MobileNetV3 [8] leveraged depthwise separable convolutions and squeeze-and-excitation modules to achieve efficient on-device inference. Transfer learning—fine-tuning ImageNet pre-trained models—has proven particularly effective when domain-specific labelled data is limited [3].

C. Sugarcane Disease Detection

Targeted research on sugarcane diseases has grown substantially in recent years. Daphal and Koli [9] proposed a multiscale attention-based approach using BSRI data, achieving competitive results. Talukder et al. [10] introduced SugarcaneNet, an ensemble of LASSO-regularised pre-trained models with improved generalisation. Kunduracioglu and Pacal [11] evaluated EfficientNet variants specifically for sugarcane classification. Yang et al. [12] proposed CaneFocus-Net with adaptive receptive fields for improved localisation. Despite these advances, most published systems lack end-to-end mobile deployment with treatment recommendation capabilities, a gap that the present work directly addresses.

III. DATASET AND METHODOLOGY

A. Dataset Description

The dataset comprises 2,521 images of sugarcane leaves categorised into five classes: Healthy, Mosaic, RedRot, Rust, and Yellow. Images were collected under varied environmental conditions—different lighting, camera angles, and backgrounds—to promote model robustness. Table I summarises the class distribution.

TABLE I
Dataset Class Distribution

	Number
Healthy	522
Mosaic	462
RedRot	518
Rust	514
Yellow	505
Total	2521

B. Data Preprocessing and Augmentation

All images were resized to 224×224 pixels to satisfy the input requirements of the selected architectures, and pixel values were normalised using ImageNet mean and standard deviation values. To mitigate overfitting and increase dataset diversity, the following augmentation techniques were applied during training: random horizontal flipping; random rotation up to ±15°; colour jitter with brightness adjustment (factor 0.2); and random zooming. The dataset was partitioned into training (80%), validation (10%), and test (10%) splits using stratified random sampling.

C. Model Architectures

Three CNN architectures were selected for comparative evaluation. Table II summarises their key characteristics.

1) ResNet-50: A 50-layer residual network using skip connections to preserve gradient flow through deep architectures. Its 25M parameters enable high-capacity feature learning, achieving superior classification accuracy at the cost of computational intensity [6].

2) EfficientNet-B0: Uses compound scaling to simultaneously optimise network depth, width, and resolution. Mobile Inverted Bottleneck Convolution (MBConv) blocks with squeeze-and-excitation modules provide 5.3M parameters with a favourable accuracy-efficiency balance [7].

3) MobileNetV3 Purpose-built for mobile and embedded deployment using depthwise separable convolutions, hard-swish activations, and squeeze-and-excitation optimisation. With only 1.5M parameters it achieves fast inference suitable for real-time mobile applications [8].

TABLE II
Model Architecture Comparison

	Parameters	Key Feature
ResNet-50	25M	Highest accuracy
EfficientNet-B0	5.3M	Balanced performance
MobileNetV3	1.5M	Fastest, mobile-friendly

D. Training Strategy

All models were initialised with ImageNet pre-trained weights and fine-tuned using a two-phase transfer learning strategy. In Phase 1 (epochs 1–10), all base layers were frozen and only the five-class classification head was trained to prevent catastrophic forgetting. In Phase 2 (epochs 11–30), selected upper layers were unfrozen and the full network was fine-tuned at a reduced learning rate.

All models were compiled with the Adam optimiser, categorical cross-entropy loss, and a batch size of 32. An initial learning rate of 1×10^{-3} was used in Phase 1 and 1×10^{-4} in Phase 2. Early stopping (patience = 5) monitored validation loss to prevent overfitting. All experiments were conducted on Google Colab with an NVIDIA Tesla T4 GPU. Table III summarises the training configuration.

TABLE III
Training Configuration

Platform	Google Colab
GPU	NVIDIA Tesla T4
Framework	PyTorch
Optimiser	Adam
Loss Function	Cross-Entropy
Phase 1 LR	1×10^{-3}
Phase 2 LR	1×10^{-4}
Batch Size	32
Max Epochs	30 (early stopping, patience=5)
Dataset Split	80% train / 10% val / 10% test

E. Evaluation Metrics

Model performance was evaluated on the held-out test set using accuracy, precision, recall, F1-score, and confusion matrices. These metrics provide complementary views of classification quality across all five disease classes. Grad-CAM visualisations were additionally generated to interpret model predictions and verify that attention was focused on disease-relevant leaf regions rather than background artefacts.

IV. SYSTEM IMPLEMENTATION

A. System Architecture

The CaneCare system follows a three-tier modular architecture: (1) a React Native mobile frontend for image acquisition and result presentation; (2) a FastAPI backend for request handling, image preprocessing, and model inference; and (3) the trained deep learning model as the classification engine. Tiers communicate via RESTful HTTP with JSON payloads.

B. Backend and API Design

The prediction API was implemented using FastAPI. The primary /predict endpoint accepts image uploads, validates

file type and size, preprocesses the image to a 224×224 normalised tensor, runs inference, and returns a structured JSON response containing: predicted disease class, confidence score, per-class probability distribution, and recommended treatment actions. A confidence threshold of 0.60 rejects ambiguous or non-leaf inputs. Additional endpoints include /health for server status monitoring and a root endpoint for API metadata. The backend is deployed on the Render cloud platform for public accessibility.

C. Mobile Application — CaneCare

The CaneCare application was developed using React Native with Expo and packaged as an Android APK via Expo Application Services (EAS). The app provides: (1) a home screen with live camera capture and gallery upload options; (2) a processing screen displaying real-time analysis feedback; and (3) a results screen showing the predicted disease, confidence percentage, severity colour indicator, all class probabilities, and step-by-step treatment recommendations. The green-themed UI is designed for intuitive use in agricultural field conditions.

D. Model Export and Deployment

Trained models were exported in PyTorch (.pth), ONNX, and TorchScript formats to support deployment flexibility. The production backend loads the TorchScript model for efficient CPU-based inference. Table IV summarises the key tools and technologies used across the full system.

TABLE IV
Tools and Technologies

Programming Language	Python
Deep Learning Framework	PyTorch
Backend API	FastAPI
Mobile Frontend	React Native (Expo)
Cloud Deployment	Render

Model Export	ONNX, TorchScript
Visualisation	Matplotlib, Grad-CAM

Yellow	0.93	0.94	0.93
--------	------	------	------

V. RESULTS AND DISCUSSION

A. Overall Model Performance

Table V summarises classification accuracy, F1-score, and parameter count for all three models evaluated on the held-out test set.

TABLE V
Model Performance Comparison on Test Set

	Accuracy (%)		
EfficientNet-B0	90.34	0.903	5.3M
ResNet-50	94.71	0.947	25M
MobileNetV3	94.18	0.941	1.5M

ResNet-50 achieved the highest classification accuracy of 94.71% with an F1-score of 0.947, benefiting from its deeper architecture and residual learning capacity. MobileNetV3 attained a closely competitive accuracy of 94.18% (F1 = 0.941) while requiring only 1.5M parameters—a 94% reduction relative to ResNet-50—and achieving faster inference (~7.2 ms vs. ~19 ms per frame). EfficientNet-B0 achieved 90.34%, reflecting a moderate trade-off between parameter efficiency and classification performance on this dataset.

B. Per-Class Performance Analysis

Table VI presents per-class precision, recall, and F1-score for the best-performing model (ResNet-50) on the test set.

TABLE VI
Per-Class Classification Metrics (ResNet-50)

	P	R	F1
Healthy	0.97	0.97	0.97
Mosaic	0.95	0.96	0.95
RedRot	0.98	0.99	0.98
Rust	0.95	0.95	0.95

RedRot achieved the highest F1-score (0.98), attributed to its visually distinctive reddish lesion patterns. Yellow Leaf Disease showed the lowest F1-score (0.93) due to its visual similarity with Mosaic disease, both producing chlorotic (yellowing) patterns. Confusion matrix analysis confirmed minor misclassification between these two classes; however, recall values exceeded 0.90 across all five classes, demonstrating strong overall generalisation.

C. Per-Class Performance Across Models

MobileNetV3 achieved the highest per-class F1-scores for Healthy (0.9744) and Mosaic (0.9565), while ResNet-50 excelled for RedRot (0.9872) and Rust (0.9554). This complementary behaviour suggests that ensemble approaches combining both models could further improve overall classification performance and represents a promising direction for future work.

D. Model Interpretability via Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) was applied to visualise the spatial regions driving model predictions. Heatmaps confirmed that the model consistently focused on disease-specific features: chlorotic stripe patterns for Mosaic and Yellow, reddish lesion edges for RedRot, and rust pustule clusters for Rust. For healthy leaves, attention was broadly distributed without concentrated focal activation. This analysis establishes that the model is leveraging biologically meaningful features rather than spurious background correlations, increasing confidence in the system's reliability for real-world deployment.

E. Discussion

The key finding is that MobileNetV3 achieves near-equivalent accuracy to ResNet-50 while being 16.7× smaller in parameter count and substantially faster in inference. This makes MobileNetV3 the recommended choice for the CaneCare mobile deployment, while ResNet-50 remains preferable for server-side scenarios where maximum accuracy is the primary objective.

The primary failure modes involve misclassification between visually similar disease classes (Mosaic vs. Yellow), reflecting a diagnostic challenge that even human experts sometimes encounter. Future work should target these classes through class-balanced augmentation and fine-grained attention mechanisms. The current limitation to image-based analysis—excluding soil, environmental, and temporal factors—also represents an avenue for extending the system toward a more comprehensive smart agriculture platform.

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented CaneCare, a deep learning-based system for automatic sugarcane leaf disease detection and classification. Three CNN architectures—EfficientNet-B0, ResNet-50, and MobileNetV3—were trained and evaluated on a 2,521-image dataset spanning five disease classes. ResNet-50 achieved the highest test accuracy of 94.71%, while MobileNetV3 attained 94.18% with only 1.5M parameters, establishing it as the preferred architecture for mobile deployment.

The complete CaneCare system integrates a FastAPI cloud backend with a React Native mobile application, enabling real-time, on-device disease predictions with confidence scores and treatment recommendations. Grad-CAM analysis confirmed that predictions are grounded in biologically meaningful disease features. The proposed system demonstrates that AI-driven precision agriculture tools can be made practical, accessible, and deployable in resource-constrained farming environments, offering a scalable pathway to reduce crop losses and support food security.

B. Future Work

Planned future directions include:

- Dataset expansion with additional disease classes, real-world field images, and diverse environmental conditions to improve model generalisation.
- Integration of disease localisation techniques (YOLO-based object detection and

segmentation) to identify and quantify disease-affected regions within leaf images.

- On-device inference via TensorFlow Lite or ONNX Runtime to enable offline prediction in areas with limited internet connectivity.
- Incorporation of IoT sensor data (temperature, humidity, soil moisture) for predictive disease outbreak modelling.
- Extension into a comprehensive smart agriculture platform encompassing yield prediction, pest detection, and fertiliser recommendation.

ACKNOWLEDGMENT

The authors thank RVS Technical Campus, Coimbatore, for providing access to computational resources and laboratory facilities that supported this research. The authors also acknowledge their project supervisor and department faculty for their guidance and encouragement throughout the development of this work.

REFERENCES

1. S. D. Daphal and S. M. Koli, "Improved multiscale attention based deep learning approach for automated sugarcane leaf disease detection using BSRI data," *Scientific Reports*, vol. 15, art. 45474, 2025.
2. X. Yang et al., "CaneFocus-Net: A sugarcane leaf disease detection model based on adaptive receptive field and multi-scale fusion," *Sensors*, vol. 25, no. 21, art. 6628, 2025.
3. S. Kumar, L. Sharma, and D. Sharma, "Hybrid deep learning framework for real-time sugarcane disease detection," in *Proc. ICDLAIR*, Atlantis Press, 2025, pp. 52–60.
4. S. Srinivasan et al., "Sugarcane leaf disease classification using deep neural network approach," *BMC Plant Biology*, vol. 25, art. 282, 2025.
5. M. S. H. Talukder et al., "SugarcaneNet: an optimized ensemble of LASSO-regularized pre-trained models for accurate sugarcane disease classification," *Journal of Big Data*, vol. 12, art. 221, 2025.

6. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE CVPR, Las Vegas, NV, USA, 2016, pp. 770–778.
7. M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in Proc. ICML, Long Beach, CA, USA, 2019, pp. 6105–6114.
8. A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF ICCV, Seoul, South Korea, 2019, pp. 1314–1324.
9. S. D. Daphal and S. M. Koli, "Enhanced deep learning technique for sugarcane leaf disease classification and mobile application integration," *Heliyon*, vol. 10, no. 8, e29438, 2024.
10. I. Kunduracioglu and I. Pacal, "Deep learning-based disease detection in sugarcane leaves: evaluating EfficientNet models," *Journal of Operations Intelligence*, vol. 2, no. 1, pp. 13–24, 2024.
11. S. Aggarwal et al., "Optimised MobileNet for very lightweight and accurate plant leaf disease detection," *Scientific Reports*, vol. 15, art. 8423, 2025.
12. A. Jain et al., "YOLO-LeafNet: a robust deep learning framework for multispecies plant disease detection with data augmentation," *Scientific Reports*, vol. 15, art. 14021, 2025.
13. P. Tipre et al., "Deep learning-based disease detection in potato and mango leaves: a comparative study of CNN, AlexNet, ResNet, and EfficientNet," *Scientific Reports*, vol. 15, art. 2788, 2025.
14. H. Sambana et al., "An efficient plant disease detection using transfer learning approach," *Scientific Reports*, vol. 15, art. 19082, 2025.
15. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in Proc. IEEE ICCV, Venice, Italy, 2017, pp. 618–626.