INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Real Time Analysis of Aloe Vera Disease Classification using Deep Learning

Narayan Sabale[1], **Sundari Tribhuvanam**[2]*, **Chandan Parasharam Kangralkar**[1], **Rahul R Halkarni**[1], **Omkar P Pote**[1]

1 Department of ECE, Jain College of Engineering, Belagavi, Karnataka, India
2 Department of ECE, Atria Institute of Technology, Bengaluru, Karnataka, India

OPEN ACCESS

Received: 05/04/2025
Accepted: 14/07/2025
Published: 29/09/2025

Citation: Sabale N, Tribhuvanam S, Kangralkar CP, Halkarni RR, Pote OP (2025) Real Time Analysis of Aloe Vera Disease Classification using Deep Learning. Indian Journal of Science and Technology 18(36): 2925-2934. https://doi.org/10.17485/IJST/v18i36.632

* **Corresponding author**.

stribhuvanam@yahoo.co.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To identify the plant diseases of *Aloe vera (A. vera)* in real time for early cure of diseases, an automatic system is designed to improve the plant yield. The objectives of this work are to develop a classification system of healthy and diseased *A. vera* leaf classes and to develop a user interface to support the real time analysis of the affected leaf diseases. **Methods:** To address the need of disease classification model, *A. vera* dataset from different sources are considered, both from open source and real time. Different deep learning models- Convolutional Neural Network (CNN), MobileNet(MNet), EfficientNet(ENet) are considered for classification of *A. vera* leaf diseases. A user friendly interface is hosted on a web platform that allows users to upload the clicked photos with their cameras of *A. vera* leaves for real time analysis and identification of diseases. **Findings:** The classification accuracy using a dataset of 9000 images for training and testing the CNN model leads to 87% accuracy, 86% precision, 91% recall, and a 0.89 F1-score. The accuracy of classification with MNet and ENet are 99% for the database considered. **Novelty:** In this work, deep learning algorithms are considered for *A. vera* leaf disease classification. A real time web based user interface provides *A. vera* growers a quick solution for the diseases found in farms by uploading the images taken using mobile phone camera and to receive remedial suggestions.

Keywords: CNN model; Deep learning; EfficientNet; Leaf disease classification; Medicinal plants; MobileNet

1 Introduction

A. vera gel is a viscous, colorless, and odorless substance with antimicrobial, antioxidant, and anti-inflammatory properties. Healthy *A. vera* leaves have a vibrant green color and a smooth, consistent texture. There are no visible signs of discoloration, spots, or damage. The leaf edges are sharp and well-defined, and the overall appearance is plump and well-hydrated. Rust disease is characterized by orange, reddish, or brown spots scattered across the leaf surface. Over time, these spots may enlarge and cause the affected areas to become dry and rough, contrasting with the healthy green sections. Leaves affected by rot show dark brown or black discoloration, often originating at the edges (Figure 1). These areas may appear soft, mushy, or slimy, often emitting a foul odor. In severe cases, the leaf experiences significant tissue breakdown [1–6].

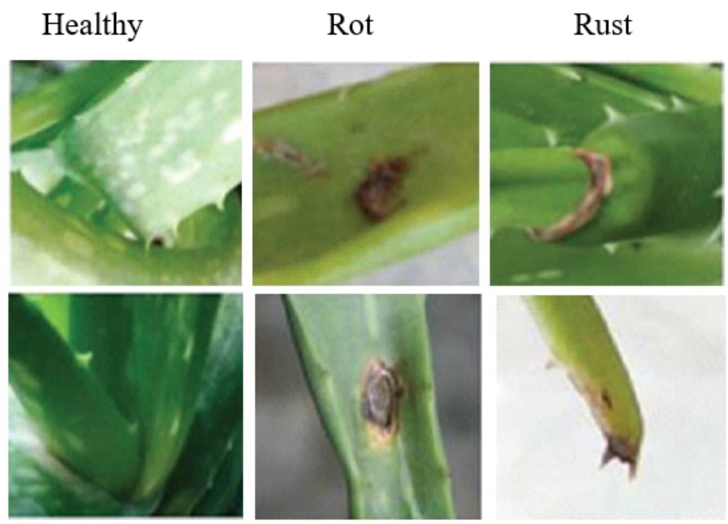


Fig 1. Samples of *A. vera* leaves (top to bottom, left to Right)

Despite their hardiness, *A. vera* plants are susceptible to various diseases that can affect both their growth and medicinal properties. Common ailments include bacterial soft rot, caused by Erwinia bacteria, leading to mushy and foul-smelling leaves; aloe rust, a fungal disease identified by brown or black spots spreading across the leaf surface due to high humidity; and leaf spot disease, characterized by small dark spots caused by fungi such as Alternaria [7]. Researchers have developed systems for leaf disease classification with machine learning techniques. Table 1 indicates earlier works of plant leaf disease identification and the associated research gaps. There is an acute need to build an automatic system for the identification of these diseases for early treatment to improve plant yield. The proposed work employs a deep learning model to examine images of *A. vera* leaves in order to assess their health status- healthy, decayed, or affected by rust. The model has been incorporated into a locally hosted website that not only identifies the leaf's condition but also suggests specific treatments based on the detected disease, ensuring proper care and management for *A. vera* plants.

2 Methodology

Methodology includes the *A. vera* plant disease database and application of effective deep learning models for classification of diseases. Subsequently, deployment of the model on the web platform for farmers to upload the captured *A. vera* leaf images from their fields for real-time analysis of the diseases. The overall flow of the work is indicated in figure 2.

2.1 Data Collection

Appropriate A. vera leaf disease dataset collection supports accuracy of classification with machine learning. The dataset used in the proposed work is sourced from the online Mendeley repository and specifically curated for classification [14]. The original A. vera disease dataset contains 2,307 images in various conditions such as Fresh, Rot, and Rust are downloaded from open source. The background-removed image dataset focuses solely on the leaf regions, enhancing disease detection accuracy.

2.2 Data Preprocessing

During preprocessing, every image of the dataset was resized to 128×128 to have image consistency of the entire database and compatibility with the model for training purposes. All the duplicate, irrelevant and corrupted images were removed during cleaning of the dataset, which eventually yielded 2,307 images of Original A. vera Disease (OAD) Dataset. The Augmented A. vera Disease Dataset is generated by applying image augmentation techniques such as flipping, rotation, noise addition, shifting, brightening, and zooming to the original dataset. This augmentation increases the database size and diversity, improving the

Table 1. Summary of earlier works and Research Gaps

References	Leaf type	Techniques Used	Data Set Used	Research Gap
(8)	Potato, Tomato, Rice	Transfer Learning, Web Deployment,	PlantVillage, Kaggle	Disease treatment not included, not interactive with users
(9)	Aloe Vera, Apple	DL, Feature Fusion, Wavelet Transform, TensorFlow	Self-collected + Augmented, PlantVillage	Focused only on Classification, no user interaction
(10)	Tomato, Apple, Rice	CNNs, Transfer Learning, GANs, Hyperspectral Imaging	PlantVillage, Kaggle, Field datasets	No user interaction
(11)	Apple, Tomato, Corn,	CNNs, Vision Transformers, GANs, XAI (LIME, SHAP), YOLOv5n	PlantVillage, AI2018, PlantDoc, Olive, CIDA	No user interaction, XAI used to aid human understanding
(12)	Banana, Custard Apple, Fig, Potato	DL for Feature Extraction, ML for Classification (Hybrid approach)	Country-specific datasets (India, Iraq, Indonesia)	Research-focused; no User interface
(13)	Tomato	ML (CV + DWT + PCA + GLCM + SVM/KNN/CNN)	Tomato Village Dataset (600 images)	Focused on modelling accuracy, no user interface.
Proposed work	Aloe Vera	CNN, MobileNet and EfficientNet for classification, Python-based Flask open-source tool (micro web framework) for user interfaces.	Mendeley (9000 images)	Classification of *A. vera* diseases. Real time analysis of *A. vera* leaf and advice to growers.

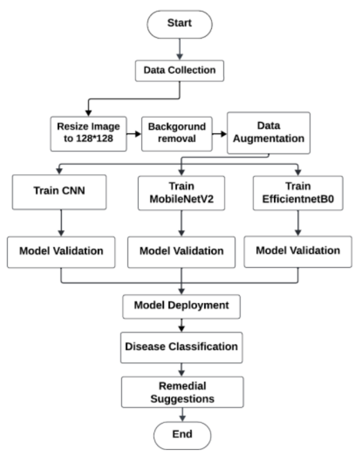


Fig 2. Overall block diagram

robustness of model training. In total, the database contains three categories: the OAD dataset with 2,307 images in JPG format, Background Removed A. vera Disease Dataset with 2,450 images in PNG format and the Augmented A. vera Disease Dataset with 9,000 images in PNG format of healthy, Rust and Rot leaf disease conditions. The typical images of the database considered in the proposed work are available at https://github.com/Sabale-37/Dataset_Aloevera.git. The data is divided into training, validation, and test sets to the tune of 70%, 20% and 10% respectively.

2.3 Model Design and Implementation

For classification of the *A. vera* diseases, different machine learning models are considered- CNN, MobileNetV2 (MNetV2) and EfficientNet-B0 (ENet-B0) for improved performance of the classifier and classification results. Convolutional Neural Network (CNN) is employed due to its capability to learn hierarchical features from images, making it ideal for classification task.

MNetV2 has more efficient CNN architecture designed for mobile and embedded vision applications. ENet-B0 is an advanced version of CNN in which higher performance can be achieved with fewer computational resources. A basic structure of the CNN model with a typical input is presented in Figure 3. The CNN architecture of the model primarily contains convolution layers, pooling layers and fully connected layers to allow the model to draw out appropriate features and perform classification and typical a three-layer model, is considered in the proposed work.

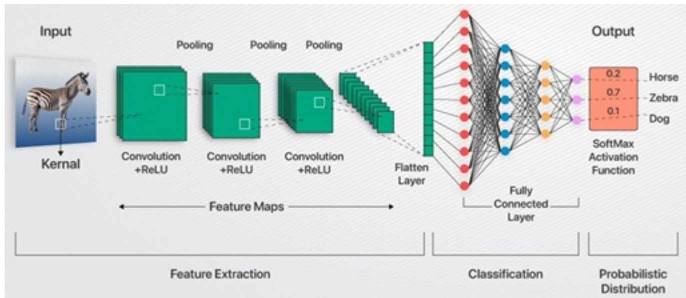


Fig 3. CNN Architecture

MNetV2 has inverted residual blocks and linear bottlenecks, resulting in higher accuracy and speed while maintaining low computational costs. **Inverted Residuals**- to connect layers with different depths, allowing for more efficient information flow and reducing computational complexity. **Linear Bottlenecks**- to reserve the information by maintaining low-dimensional representations, which minimizes information loss and improves the overall accuracy of the model. MNetV2 architecture has **Depthwise Separable Convolutions** to reduce the number of parameters and computations. This technique splits the convolution into two separate operations: depthwise convolution and pointwise convolution, significantly reducing computational cost. **ReLU6 Activation Function** clips the ReLU output at 6 to prevent the numerical instability in low-precision computations, making the model more suitable for mobile and embedded devices. The architecture has input layer that takes input of RGB images of size 224x224. Initial convolution layer is with stride 2. Inverted Residual Blocks have an expansion layer, depthwise convolution, and a projection layer. The last layers are Conv2D Layer, Average Pooling Layer and Fully Connected Layer[(15)].

The ENet-B0 has three major parts. (1) Stem is the first layer with convolution with 32 filters with kernel size 3×3 and stride 2 followed by batch normalization and a ReLU6 activation function. (2) Body consists of a series of MBConv blocks with configurations as given in Table 2. Each block consists of depthwise separable convolutions and squeeze-and-excitation (SE) layers. Typical MBConv block parameters are: Expansion ratio-the factor by which the input channels are expanded, Kernel Size, Stride and SE Ratio. (3) Head-includes final convolution followed by global average pooling layer and a fully connected layer with softmax activation function for classification. ENet-B0 is the baseline model without any scaling and designed to give moderate depth, width, and resolution[(16)]. The hyper parameter setting of CNN, typical architectures of MNetV2 and ENet-B0 is given in figure 4.

2.4 Training the machine learning Models

The dataset consisted of images categorized into three classes: Fresh, Rot, and Rust, representing different A. vera leaf conditions. The database is a balanced set of all three classes with 3000 images of each class. The models were trained, tested on a validation dataset to assess its performance in accurately classifying the diseases based on the input images.

2.4.1. Training with CNN Model:

The preprocessed dataset is fed to train the three-stage CNN model with hyper parameter setting as indicated in Table 2 with 124×124 size input images.

2.4.2. Training with MobileNet Model:

The image dataset is resized to 224×224 as per the MNetV2 model configuration and normalized. The MNetV2 is loaded with pretrained weights from ImageNet excluding the fully connected top layers to customize for 3-class classification task. The base model is freezed to retain learned features and to avoid overfitting. Global average pooling layer is added to reduce model complexity. Dense layer with softmax activation is added for the output classes. The model is compiled with categorical cross-entropy as the loss function and accuracy as the evaluation metric. The model is trained using Adam optimizer on training

Hyper parameter setting for CNN		
Layer	Output	Hyperparameters
Input	128x128x1	
Conv_1	128x128x64	F=64; K=5x5; S=1;
Max-POOL_1	64x64x64	Pool=2x2;S=2
Conv_2	64x64x64	F=64; K=5x5; S=1;
Max-POOL_2	32x32x64	Pool=2x2;S=2
Conv_3	32x32x64	F=64; K=5x5; S=1;
Max-POOL_3	10x10x64	Pool=2x2;S=2
Flatten	3200	
FC1	256	Neurons=256
FC2	128	Neurons=128
CNN OUTPUT	1	Optm=ADAM Loss=binary_Crossentropy
F:depth of Conv layers K:kernel size S:stride		

Typical architecture of MobileNetV2					
Input Size	Operator	Expansion Factor (t)	Output Channels(c)	Number of Repeats (n)	Stride (s)
224x224x3	Conv2D	-	32	1	2
112x112x32	Bottleneck	1	16	1	1
112x112x16	Bottleneck	6	24	2	2
56x56x24	Bottleneck	6	32	3	2
28x28x32	Bottleneck	6	64	4	2
14x14x64	Bottleneck	6	96	3	1
14x14x96	Bottleneck	6	160	3	2
7x7x160	Bottleneck	6	320	1	1
7x7x320	Conv2D	-	1280	1	1
7x7x1280	AvgPool 7x7	-	-	1	-
1x1x1280	Conv2D	-	k(number of classes)	1	-

Typical architecture of EfficientNet-B0				
Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

Fig 4. Typical architectures of CNN, MNetV2 and ENet-B0

data for ten epochs. The last few layers are unfreezed to perform fine-tuning to allow the model to adjust high-level features for the training dataset while retaining its foundational knowledge. The model is evaluated with test set with loss and accuracy measurement.

2.4.3. Training with EfficientNet Model:

ENet-B0 architecture with transfer learning is to create a classification system for the detection of *A. vera* leaf diseases. Every image was resized to 224 × 224 pixels and normalized. A five-fold cross-validation approach was used to guarantee a strong assessment of the model. The base model was supplemented with custom classification layers such as Global Average Pooling, Dense, and Dropout layers. To enable thorough fine-tuning on the target dataset, all layers of ENet-B0 were unfrozen, in contrast to feature extraction techniques that maintain the base layers frozen. The model was trained for 10 epochs per fold with a batch size of 32 using the Adam optimizer with a learning rate of 0.0005.

2.5 Model Evaluation

After training, the models were validated with validation dataset. The models were tested individually with test dataset with 600 images of each class to compute the performance metrics as described in the following paragraph.

Accuracy: This metric indicates the proportion of correctly predicted samples compared to the total number of samples. It is widely used to assess the overall effectiveness of a classification model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall: Recall measures the proportion of actual positive cases correctly identified by the CNN model. It becomes especially important when the cost of missing true positive cases (false negatives) is significant, such as in the detection of rare diseases.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Precision: Precision represents the fraction of positive predictions that are actually correct. It is especially important when the cost of false positives is significant.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

F1-Score: The F1-score is the harmonic mean of precision and recall and indicates the balance of both. F1-score is highly valuable when the class is imbalanced because this makes sure both precision and recall are considered for assessing the performance.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

These metrics play a key role in assessing the performance of the classification model, particularly when dealing with imbalanced datasets. The confusion matrix identifies the strengths and limitations of the model.

2.6 Model Deployment

The trained model is deployed on a web platform for user interaction and for real-time analysis of the *A. vera* leaves. This user friendly interface allows users to access the classification functionality conveniently. The user can interact with the website by uploading the captured images of *A. vera* leaves. A web app is designed and released for user interaction. The interface is simple and lightweight but with strong image classification features. HTML, CSS, and JavaScript were used to build the front end for easy usage and interaction. The open-source Flask web framework is used for the backend to handle image uploads and run machine learning inferences. The app is hosted on the Render cloud platform. Users can upload images of *A. vera* leaves by drag and drop technique or by manual upload. The homepage has a simple, clean design. The trained CNN model tests the uploaded image and classifies the leaf into one of three categories: Healthy, Rust, or Rot. After classifying, the app shows the result of classification along with required suggestions to the users. The suggestions include disease symptoms, possible causes, and suggested treatments. The interface is demonstrated with test images at https://aloevera-disease-classification-using-cnn-3og7.onrender.com. The proposed work is an all-in-one solution for the *A. vera* growers with an AI-based real-time and accurate disease identification system, empowering users to take quick, timely actions to protect the healthy *A. vera* yield.

3 Results and Discussion

The CNN-based model correctly categorizes *A. vera* leaves as Fresh, Rot, or Rust based on the input image. A total of 600 images from each class are used to test the model and to evaluate the performance metrics of the models. Figure 4 illustrates the confusion matrices of the models considered to provide meaningful insights into model accuracy, precision, recall and the distribution of correct and incorrect predictions for all classes. Analysis of the Confusion Matrix of CNN indicates, out of 600 sample of each category, 546 healthy, 504 Rot and 522 Rust samples classified correctly leading to accuracies of 91%, 84% and 87% for healthy, Rot and Rust categories respectively. The precision, recall and F1-score of CNN are indicated in Table 2. Similarly, the performance metrics of MNetV2 and ENet-B0 are also tabulated in **Table 2,** indicates the accuracy of 99% and F1-score of 0.99 indicates correctness in classification for the dataset under consideration. Confusion Matrices generated are given in f**igure 5**.

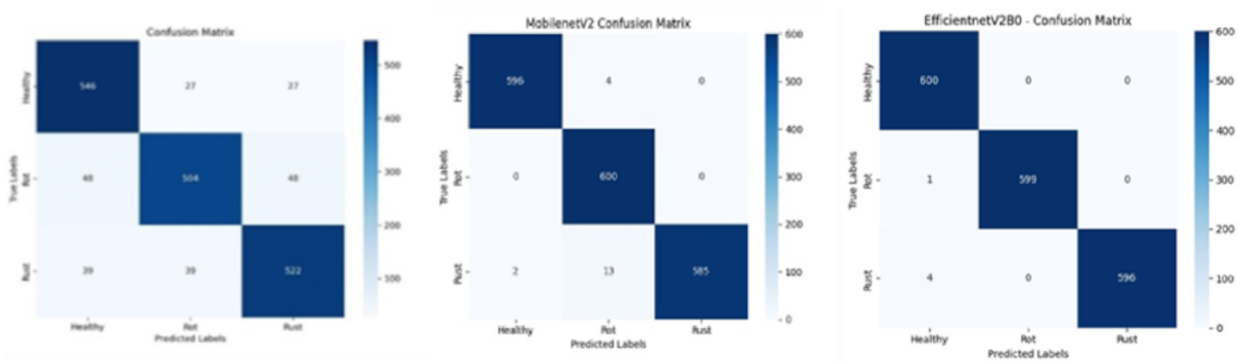
Table 2. Classification Reports of CNN, MNetV2,ENet-B0

Model	CNN			MobileNetV2			EfficientNetB0			Support
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
Healthy	0.86	0.91	0.89	1.00	0.99	0.99	1.00	0.99	0.99	600
Rot	0.88	0.84	0.86	0.97	1.00	0.99	0.97	1.00	0.99	600
Rust	0.87	0.87	0.87	1.00	0.97	0.99	1.00	0.97	0.99	600
Accuracy			0.87			0.99			0.99	1800
Macro avg	0.87	0.87	0.87	0.99	0.99	0.99	0.99	0.99	0.99	1800
Weighted Avg	0.87	0.87	0.87	0.99	0.99	0.99	0.99	0.99	0.99	1800

Macro Average: The average of precision, recall, and F1-score across all classes without considering the sample size is 0.87. *Weighted Average*: Takes the support (sample size of each class) into account when averaging metrics, ensuring a more balanced evaluation is 0.87. Equal values of Macro Average and Weighted Average indicate the appropriateness of the model training and test dataset. Both MNetV2 and ENet-B0 give same results for classification indicates the models are interchangeable depending on the resource availability. The dataset under consideration is small in size and results clearly indicate the models can handle much bigger dataset by fine tuning of the parameters of the models.

Figure 6 presents the Receiver operating characteristic (AUC-ROC) curves of the models, which charts the True Positive Rate (TPR) against the False Positive Rate (FPR). This curve visually demonstrates how well the model can differentiate between the various classes. A higher TPR and lower FPR reflect a more effective model. The ROC metric offers a quantitative evaluation of the model's ability to classify instances correctly. A ROC value approaching 1 signifies a model with high discriminative power. Typical results of classification with ENet-B0 are presented in Figure 7.

The result of user interface deployment is shown in figure 8, requesting the user to upload an *A. vera* leaf image captured by camera on the web platform. The interface is user-friendly, displaying the predicted disease class alongside specific



(a) CNN (b) MobileNetV2 (c) EfficientNet-B0

Fig 5. Confusion Matrices of classification

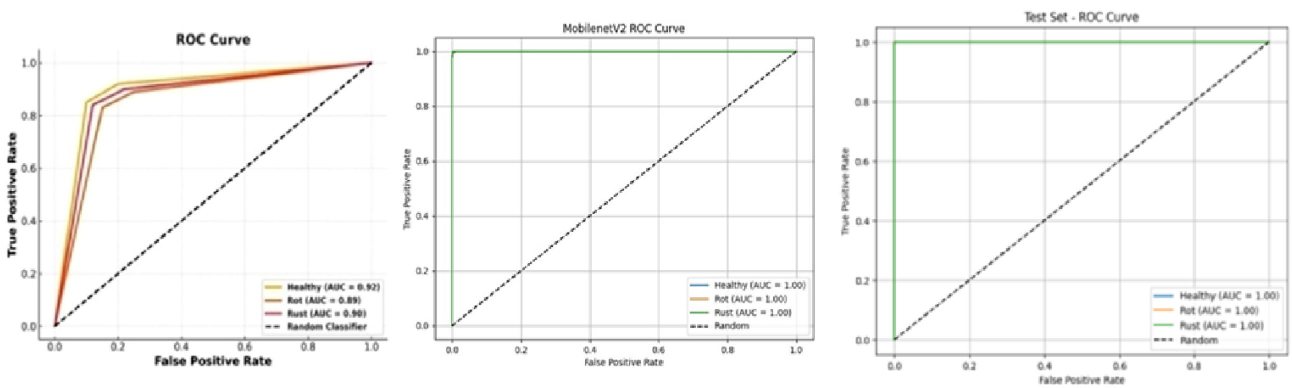


Fig 6. ROC Curves for CNN, MNetV2, ENet-B0

Fig 7. Typical test output of EfficientNet-B0

treatments/suggestions for the identified disease. In this typical case, the model has classified the input image as "Aloe Vera Rust Disease" and provides corresponding treatment suggestions, such as cutting off infected leaves, to maintain proper air circulation, to apply sulfur or copper-based fungicide, and to avoid over watering. This visual output aids the growers in consulting local experts to identify plant diseases and accurately take appropriate remedial actions.

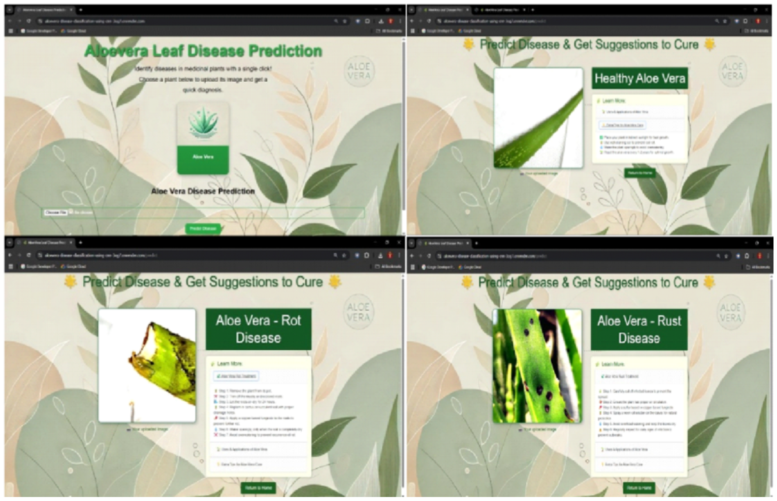


Fig 8. Output interface for user image upload and suggestions to user

The deep learning models used in this study have achieved classification accuracy upto 99%, showcasing reliability and effectiveness in distinguishing healthy leaves from diseased ones. In comparison with earlier research, the proposed work gives significant accuracy and web based user interface supports the rural farmers with a handy tool to monitor the quality of the *A. vera* yield continuously. Earlier works focus on different species of plant leaves based on available datasets, and our work proposes an exclusive *A. vera* leaf disease prediction system with real time user interface. Table 3 summarizes the outcomes of the similar existing work in comparison with the proposed classification model.

Table 3. Comparison of outcomes of the related works

References	Accuracy	Real time UI	Speed	User Engagement
Emma Harte (2020) [8]	97.2% (Validation), 44% (In-field)	No	Slow for in-field use	No interactivity
Nazeer Muhammad et al. (2021) [9]	Aloe: 96.6%, Apple: 99.1%	No	Moderate	Disease Classification only
Saurav Sagar et al. (2023) [11]	Up to 99.94%	Partial	Fast with YOLOv5n	XAI used to aid human understanding
Sujatha et al. (2025) [12]	Up to 99.1% (CA), 91.9% (Banana), 62.6% (Potato), 86.5% (Fig)	No	Moderate	Research-focused; no UI
Sunil Harakannanavar et al. (2022) [13]	CNN: 99.09%, SVM: 89%, KNN: 97.3%	No	Moderate	Focused only on accuracy of the system
Malek, M.A. et al. (2023) [17]	Deep learning model upto 94.11%	No	moderate	Focused only on accuracy of classification
K. Saini et al. (2024) [18]	Deep learning:96.59%	No	Moderate	Focused only on accuracy of classification
Proposed method	CNN Aloe 87% MobileNet 98% EfficientNet 99%	Yes	Fast(<1sec) due to Flask API	Real-time disease analysis, Interactive with suggestions to users

The challenge is to find an appropriate light weight model for user interface. Expanding the training dataset could further improve classification performance. The data augmentation techniques such as rotation, flipping the images not only expand the dataset, but also allow the model to better adapt to real-world variations, such as changes in angle and lighting while capturing image in a camera. However, there were some challenges that were noted, particularly regarding lighting variations and class imbalance, where healthy leaf images were more prevalent than diseased ones. The free, open source render platform introduces a latency of 1 sec. Latency can be reduced by considering paid version of render platform. Mobile phone with Android application and good internet connection is the basic requirement to implement this work. Low bandwidth setting poses a limitation for real time analysis and the farmer should adopt to non real time analysis to identify the *A. vera* leaf disease class. Future work will involve gathering a more balanced dataset and introducing more sophisticated augmentation strategies to deal with the identified problems. This work can be extended for other crops also supporting a large farming community. Overall, the model's significant accuracy and dependable disease detection capabilities confirm its potentiality for applied use in agricultural monitoring for the early identification and timely intervention of *A. vera* plant health management.

4 Conclusion

This study proposes an AI based disease detection and classification of healthy and unhealthy *A. vera* leaves. Healthy, Rot and Rust disease classes of *A. vera* were considered. The primary focus of this work is classification with deep feature selection of the curated *A. vera* database. The Mendeley *A. vera* database was considered and image augmentation techniques were applied to enhance the images. Preprocessing techniques such as flipping, rotation, and background removal were considered to expand the database size. A three stage CNN model, MobileNetV2 and advanced EfficientNet-B0 were implemented for classifying *A. vera* plant diseases, achieving upto 99% accuracy in detecting Fresh, Rot, and Rust conditions. The models demonstrated significant performance with reliable performance metrics. By utilizing augmented images, the model effectively handles diverse conditions, but future improvements will focus on addressing class imbalance and lighting variations. A web based user interface is developed to facilitate *A. vera* growers to upload the images of suspected diseases for better identification purpose in real time. The interface also provided users of the proposed system with suggestions, possible precaution and treatment to support timely consultation with local experts to protect the *A. vera* yield. This work contributes to the advancement in plant disease detection, offering valuable insights for sustainable agriculture and plant health management.

5 Acknowledgements

The research is an outcome of internship carried at ECE Research Centre, Atria Institute of Technology, Bengaluru. Authors are thankful for IEEE CAS, Bangalore section and Principal Atria IT for providing an opportunity to carryout internship at Atria IT, Bengaluru, India

6 Contributory Statement

The contributions of each author to this research are as follows:

1. Conceptualization of Study Problem: Dr. Sundari Tribhuvanam
2. Software/Algorithm Development: Narayan Sabale
3. Data Collection and Preprocessing: Chandan Parasharam Kangralkar
4. Original Drafting: Omkar Pote
5. Experimental Design: Rahul R Halkarni
6. Editing and Reviewing: Dr. Sundari Tribhuvanam

Each author has significantly contributed to the respective sections of the research, ensuring the integrity and quality of the study.

References

1) Prisa D. Aloe: medicinal properties and botanical characteristics. *Journal of Current Science and Technology*. 2022;12(3):605–614.
2) Avasthi S, Gautam AK, Bhadauria R. First report of leaf spot disease of Aloe vera caused by Fusarium proliferatum in India. *Journal of Plant Protection Research*. 2023;58(2):109–114. https://doi.org/10.24425/119125.
3) Pandey A, Pande SK, Kumar K, Vishwanath HS, Kumar S. Epidemiology of Leaf Spot Disease of Aloe Vera. *International Journal of Current Microbiology and Applied Sciences*. 2020;Special Issue-11:869–872. Available from: https://www.researchgate.net/publication/353480151_Epidemiology_of_Leaf_Spot_Disease_of_Aloe_vera.

4) Mitra A, Singh M, Banga A, Pandey J, Tripathi SS, Devraj. Bioactive Compounds and therapeutic properties of Aloe vera-A Review. *Plant Science Today*. 2022. https://doi.org/10.14719/pst.1715.
5) SaVanna Shoemaker. How to make Aloe Vera Gel. *Healthline Nutrition*. Available from: https://www.healthline.com/nutrition/how-to-make-aloe-vera-gel#what-you-need.
6) Liang J, Cui L, Li J, Guan S, Zhang K, Li J. Aloe vera: A Medicinal Plant Used in Skin Wound Healing. *Tissue Engineering, Part B: Reviews*. 2021;27(5):455–474. https://doi.org/10.1089/ten.teb.2020.0236.
7) Ayodele SM, Ilondu EM. Fungi associated with base rot disease of Aloe vera (Aloe barbadensis). *African Journal of Biotechnology*. 2008;7(24):4471–4474. Available from: https://academicjournals.org/journal/AJB/article-full-text-pdf/F5B44AB8965.
8) Harte E. Plant Disease Detection using CNN. 2020. https://doi.org/10.13140/RG.2.2.36485.99048.
9) Muhammad N, Rubab, Bibi N, Song OY, Khan MA, Khan SA. Severity Recognition of Aloe vera Diseases using AI in Tensor Flow Domain. *Computers Materials Continua*. 2021;66(2):2199–2216. https://doi.org/10.32604/cmc.2020.012257.
10) Li L, Zhang S, Wang B. Plant Disease Detection and Classification by Deep Learning-A Review. *IEEE Access*. 2021;9:56683–56698. https://doi.org/10.1109/ACCESS.2021.3069646.
11) Sagar S, Javed M, Doermann DS. Leaf-Based Plant Disease Detection and Explainable AI. *Computer Vision and Pattern Recognition*. 2023. https://doi.org/10.48550/arXiv.2404.16833.
12) Sujatha R, Krishnan S, Chatterjee JM, Gandomi AH. Advancing Plant Leaf Disease Detection Integrating Machine and Deep Learning. *Scientific Reports*. 2025;15:11552. https://doi.org/10.1038/s41598-024-72197-2.
13) Harakannanavar S, Rudagi JM, Veena I, Puranikmath VI, Siddiqua A. Plant Leaf Disease Detection using Computer Vision and Machine Learning Algorithms. *Global Transitions Proceedings*. 2022;3:305–310. https://doi.org/10.1016/j.gltp.2022.03.016.
14) Nirob MAS, Praymma B, Khatun T, Uddin MS. Aloe Vera Diseases Dataset for Classification of Aloe Vera Diseases Using Machine Learning. Version 2 ed.. 2024. Available from: https://data.mendeley.com/datasets/cksmdjw8gy/2. https://doi.org/10.17632/cksmdjw8gy.2.
15) Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018;p. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.
16) Agarwal V. Complete Architectural Details of all EfficientNet Models. *Medium*. 2020. Available from: https://medium.com/data-science/complete-architectural-details-of-all-efficientnet-models-5fd5b736142.
17) Malek MA, Debnath A, Reya SS. Classification of Aloe Vera Leaf Diseases Using Deep Learning. Singapore. Springer. 2023. https://doi.org/10.1007/978-981-99-8937-9_40.
18) Saini K, Agarwal P. Aloe Vera Disease Prediction using Machine Learning Techniques. *5th International Conference on Smart Electronics and Communication (ICOSEC)*. 2024;p. 1117–1122. https://doi.org/10.1109/ICOSEC61587.2024.10722720.