IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

AI-POWERED ARECANUT PLANT DISEASE DIAGNOSIS AND KANNADA LANGUAGE CHATBOT FOR PRECISION AGRICULTURE

Dr. Deepa V B, M.Tech., Ph.D.[1], S G Santosh[2], Varun V[3], Vinayaka N Naikar[4], Lohith BS[5]

Assistant Professor, Department of Information Science and Engineering, Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India – 577201[1]

Students, Department of Information Science and Engineering, Jawaharlal Nehru New College of Engineering, Shivamogga, Karnataka, India – 577201[2,3,4,5]

Abstract: Arecanut is an important plantation crop in Karnataka, and delayed recognition of plant-health problems can reduce yield and farm income. Manual inspection is subjective and time-consuming, and technical crop-health information is rarely available in the regional language. This paper presents a preliminary decision-support system that couples deep-learning image classification with a Kannada-language chatbot interface for arecanut farmers. An uploaded image is resized to 227 × 227 pixels, rescaled, and classified by a sequential convolutional neural network (CNN) with an AlexNet-style arrangement of five convolutional layers and batch normalization, trained with the Adam optimizer and rotation, zoom and horizontal-flip augmentation for 50 epochs. A Flask web application returns the predicted class with related symptom and management information and hosts the "Areca Assistant" chatbot, which answers follow-up questions in Kannada. After 50 epochs the model reached a training accuracy of 99.03% and a validation accuracy of 99.67%, and a scikit-learn classification report on 8,143 evaluation samples gave macro-averaged precision, recall and F1-score of 0.98, 0.99 and 0.98. The labelled data span several crops, and an independent test set, field trials and a quantitative chatbot evaluation remain to be carried out. The main contribution is the integration of image-based screening and regional-language assistance in a single workflow that can support smallholder arecanut growers in early disease awareness.

Keywords: Arecanut Disease Detection, Deep Learning, Convolutional Neural Network, Image Classification, Natural Language Processing, Kannada Chatbot, Precision Agriculture, Smart Farming

I. INTRODUCTION

This work lies in the domain of artificial intelligence, computer vision and natural language processing applied to agriculture. Agriculture remains an important source of livelihood in India, and plantation crops support many rural households. Arecanut (*Areca catechu* L.), also called betel nut, is cultivated widely in Karnataka, and farmers in the coastal and Malnad regions depend on healthy palms for a regular harvest.

Disease, nutritional problems, weather conditions and management practices all influence the health and productivity of an arecanut plantation. Early identification of such problems is important because delayed recognition allows a problem to spread or worsen. Traditional inspection relies on visual observation and the farmer's experience. Although experienced farmers recognize many symptoms, symptoms of different disorders overlap and are difficult to classify consistently, and regular field inspection takes considerable time when a farmer manages a large area. A farmer may also have to travel to an expert or rely on informal advice to interpret what is seen on the leaves.

Artificial intelligence and computer vision can assist with image-based screening. A convolutional neural network (CNN) learns visual patterns such as colour change, spots and texture from labelled leaf images and uses them to predict a class for a new image. In a practical application a farmer can capture or select a leaf image, upload it to a web application and receive a predicted class together with a short explanation rather than a bare technical label. CNNs have been applied to arecanut disease detection in several studies [1]–[3], and transfer-learning models have been compared for the same task [4], [5].

Language accessibility is equally important. A prediction is less useful if the farmer cannot readily understand it or ask what to do next, and technical crop-health information is frequently written in English, which is a barrier for Kannada-

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511152

speaking growers. A chatbot in Kannada offers a conversational way to ask about the detected class, visible symptoms and basic preventive practices. Multilingual advisory systems and voice assistants for farmers have been reported [6]–[8], but they are generally separate from image-based diagnosis of arecanut.

The proposed system is designed as a preliminary decision-support tool and not as a replacement for expert diagnosis. Its predictions depend on the quality and variety of the training images and may be less reliable for unusual symptoms, poor lighting, multiple simultaneous diseases or images that differ from the training data. The application is therefore intended to communicate this limitation and to encourage confirmation by an agricultural officer when symptoms are severe or unclear.

The objectives of the work are: (i) to collect and preprocess labelled plant images for training; (ii) to design a deep-learning model that classifies the images; (iii) to evaluate the model with appropriate performance measures; and (iv) to integrate a Kannada-language chatbot that supports farmers with disease-related information. The main contributions are an AlexNet-style sequential CNN with a defined preprocessing and augmentation pipeline, a web application that returns the predicted class with symptom and management information, an integrated Kannada chatbot panel, and a transparent account of what has and has not yet been evaluated.

The rest of the paper is organized as follows. Section II reviews related works, Section III describes the system architecture, Section IV details the implementation, Section V presents experimental results and analysis, Section VI concludes the paper, and Section VII outlines future scope.

II. RELATED WORKS

A. CNN-Based Arecanut Disease Detection

Balipa *et al.* [1] used a CNN to extract features from arecanut leaf images and an SVM to classify the leaves as healthy or diseased, with the pipeline comprising image collection, preprocessing, feature extraction and classification. Madhu *et al.* [2] used a CNN on images of healthy and diseased leaves collected from farms and described the automatic extraction of colour, texture and disease-pattern features. Meghana and Prabhudeva [3] combined image-processing steps such as resizing, noise removal, filtering and enhancement with a CNN. The limitations commonly noted for these approaches are the need for large labelled datasets, high computational cost for training, and sensitivity to lighting, image quality and background conditions.

B. Transfer Learning and Comparative Studies

Veeresha *et al.* [4] trained a ResNet-50 CNN to separate healthy leaves from leaves affected by yellow leaf disease (YLD), which is a major arecanut disease; their model therefore covers a single disease and may not detect other leaf diseases. Beena *et al.* [5] trained and compared MobileNetV2, ResNet and VGG-16 for arecanut disease prediction, where MobileNetV2 offers lightweight and faster processing, ResNet improves deep feature learning and VGG-16 gives strong image classification, at the price of higher computational requirements and more complex implementation when several frameworks are used.

C. Remote Sensing, Automation and Recommendation Systems

Lei *et al.* [9] detected YLD in arecanut plantations with UAV-mounted RGB, thermal and multispectral sensors, which allows large areas to be monitored quickly but requires expensive equipment and trained operators and depends on weather. Ashritha *et al.* [10] combined CNN-based arecanut disease detection with cluster-based farm management and an automated pesticide-spraying mechanism to reduce chemical use; the need for drones, sensors and spraying units makes such a system costly and complex to integrate. Gupta *et al.* [11] proposed a generative-AI system that analyses arecanut leaf pathology and generates recommendations such as preventive measures and pesticide use. Such recommendation systems go beyond classification, but they carry the risk of incorrect or poorly grounded advice if the underlying model is not well trained or updated.

D. Language-Assisted Agricultural Tools

Amulya and Lakshmi [6] combined CNN-based crop-disease detection with multilingual guidance on pesticide usage, prevention and crop care, although for crops in general and not arecanut specifically; they note that translation quality can affect the recommendations. Palepu *et al.* [7] describe Agriguide, a voice assistant that uses speech recognition and natural language processing to answer farmers' spoken queries in their native language, and Anjani *et al.* [8] describe a Kannada speech-recognition enquiry system that retrieves answers from a stored knowledge base. Both depend on internet connectivity and on a limited knowledge base. Kannada language technology also includes a Transformer-based speech-to-text model [12] and a GRU encoder–decoder with hybrid attention for English-to-Kannada translation [13].

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

General surveys of natural language processing [14], [15] describe the text-processing steps (tokenization, stop-word removal, stemming, lemmatization) and the machine-learning methods that underlie chatbots and virtual assistants, and note that performance can degrade for low-resource languages and specialized domains.

E. Research Gap and Position of the Proposed Work

From the literature reviewed here, the arecanut studies [1]–[5] concentrate on image classification, often with two classes or a single disease, and are presented as models rather than farmer-facing tools. Remote-sensing and automation approaches [9], [10] need costly equipment, and the recommendation and multilingual systems [6], [11] do not offer a Kannada conversational interface for arecanut. Kannada speech and translation work [8], [12], [13] is not linked to image diagnosis. The present work addresses this gap by placing an image classifier and a Kannada chatbot in one user workflow, using only a web browser. Because the cited studies use different datasets and protocols, a numerical comparison with them is not attempted.

III. SYSTEM ARCHITECTURE

The proposed system consists of two principal components, an image-based arecanut leaf classifier and a Kannada-language chatbot, connected through a web interface that presents the output to the user. The architecture is modular, so that the classifier, the application server and the chatbot can be improved independently, as depicted in Fig. 3.1.

A. Overall System Architecture

The farmer uploads a leaf image through the web interface. The application server validates the file and passes it to the preprocessing stage, which resizes and rescales the image. The resulting tensor is given to the trained CNN, which returns class scores; the class with the highest probability is mapped to a readable label, and the result page displays the label with associated symptom and management information. The chatbot panel on the same page accepts follow-up questions and returns responses in Kannada, optionally using the predicted class as context. The block that produces the chatbot responses is labelled “Kannada NLP / response module” in Fig. 3.1 because its exact technique must be stated by the authors (Section 3.8).

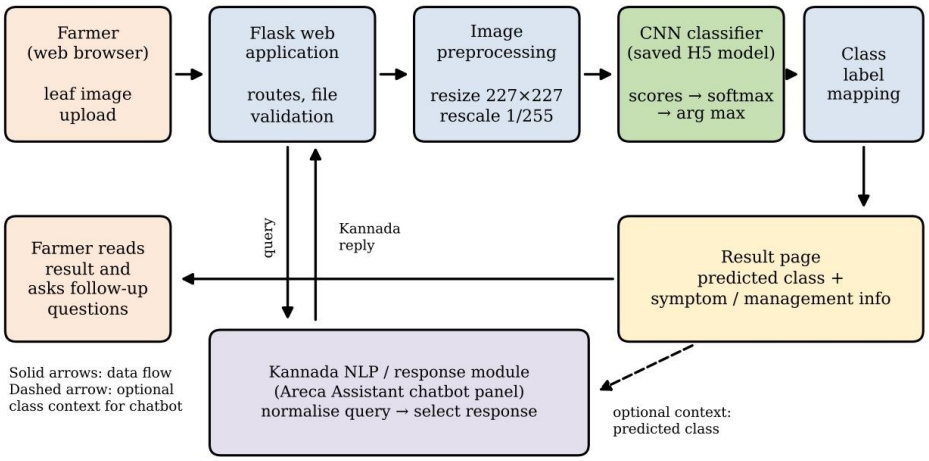


Fig. 3 1 Proposed system architecture (drawn from the system description in the project report).

The end-to-end workflow is summarized in Fig. 3.2. The flowchart is drawn as a healthy/diseased decision for simplicity; if a disease is detected, the disease type and treatment suggestions are displayed, otherwise the leaf is reported as healthy, and the suggestion is presented in the regional language. The implemented classifier is multi-class, as described in Sections 3.4 and 3.5.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

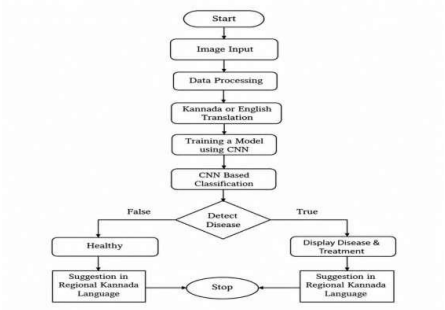


Fig. 3 2 System workflow diagram (from the project report).

B. Image Acquisition

Input to the system is a photograph of an arecanut leaf or plant part, captured with a camera or mobile phone or selected from stored files, and uploaded through a file chooser on the web page. The interface can be opened on a computer or on a compatible mobile browser, depending on the deployment environment. Prediction quality depends on image quality: lighting, leaf orientation, background, camera quality and disease severity all influence what the model sees, and training data should represent this variation as far as possible.

C. Image Preprocessing and Augmentation

The training configuration uses an input size of 227 × 227 pixels and a batch size of 32. Images are rescaled by 1/255, so that the preprocessing function P maps an uploaded image X to the network input X′, as in (1). Training images are further augmented by random rotation, zoom and horizontal flipping, which exposes the model to modest variations in orientation and framing. Validation images are only rescaled, without random augmentation, so that validation measurements remain consistent.

$$X' = P(X) = \text{Resize}(X) / 255 \qquad (1)$$

At inference the uploaded image must undergo the same processing: it is read, converted to the required colour format, resized, rescaled and given a batch dimension. Using a different preprocessing method at inference than in training, for example the preprocessing function of another backbone, would reduce prediction reliability. The report also describes colour-space (HSV) analysis and the percentage of yellow and brown pixels as a possible complementary step; it was not evaluated and is not part of the pipeline reported here.

D. Deep Learning Model

A CNN is suitable for image classification because convolutional layers learn local visual features such as edges, textures, spots, colour changes and lesion patterns; pooling layers reduce spatial dimensions, and batch normalization improves training stability. AlexNet [16] contains five convolutional and three fully connected layers. The model summary captured from the training run (Fig. 3.3) shows a sequential Keras network whose five convolutional layers follow the AlexNet filter configuration (96, 256, 384, 384 and 256 filters) and spatial sizes (55 × 55, 27 × 27 and 13 × 13), with batch normalization after the first two convolutions. The implemented classifier is therefore described as an AlexNet-style sequential CNN, not as the original AlexNet. Table 3.1 lists the layers visible in the summary; kernel sizes and strides are not displayed and are inferred from the parameter counts and output shapes (11 × 11 with stride 4, then 5 × 5). The reference design in the report uses ReLU, f(x) = max(0, x), and fully connected layers before the softmax output; the activations, dense layers and dropout of the implemented model must be confirmed from the training code.

TABLE III 1 LAYERS VISIBLE IN THE KERAS MODEL SUMMARY (INPUT 227 × 227 × 3)

Layer	Output shape	Parameters
Conv2D (96 filters)	55 × 55 × 96	34,944
BatchNormalization	55 × 55 × 96	384
MaxPooling2D	27 × 27 × 96	0
Conv2D (256 filters)	27 × 27 × 256	614,656
BatchNormalization	27 × 27 × 256	1,024
MaxPooling2D	13 × 13 × 256	0
Conv2D (384 filters)	13 × 13 × 384	885,120
Conv2D (384 filters)	13 × 13 × 384	1,327,488

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

Layer	Output shape	Parameters
Conv2D (256 filters)	13 × 13 × 256	884,992

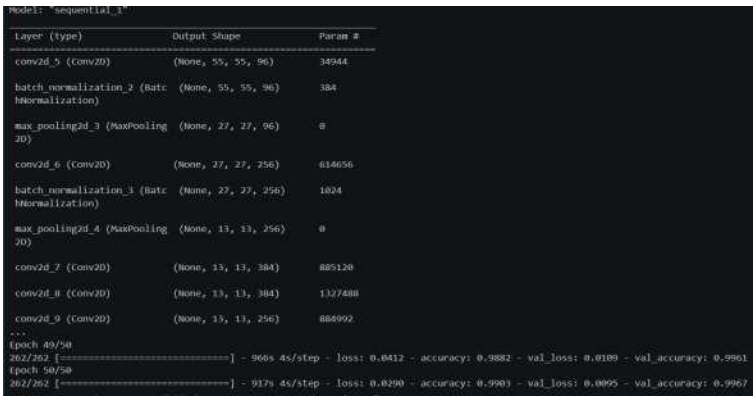


Fig. 3 3 Partial Keras model summary and the final two training epochs, captured from the training run.

E. Disease Classification

For K classes the network produces the score vector in (2), which the softmax layer converts to class probabilities in (3). The predicted class is the index of the largest probability (4), and it is mapped to a label using the same class order that was used in training; otherwise a correct model output could be displayed under the wrong disease name. The probabilities are model scores and should not be read as calibrated real-world probabilities of disease. Training minimizes the categorical cross-entropy in (5), the usual objective for a softmax output.

$$z = f_\theta(X') = [z_1, \ldots, z_K] \tag{2}$$

$$p_i = \exp(z_i) / (\sum_{j=1}^{K} \exp(z_j)) \tag{3}$$

$$\hat{y} = \arg\max_i p_i \tag{4}$$

$$L(y, p) = -\sum_{i=1}^{K} y_i \log p_i \tag{5}$$

The disease categories shown by the application must match the labels actually used to train the model. Examples of arecanut problems discussed in the project context, such as YLD, Koleroga (Mahali) and leaf-spot (Anaberoga) symptoms, should be treated as output classes only if they are present in the model's class mapping.

F. Backend and Frontend Integration

A Flask application acts as the application server and connects image upload, preprocessing, model inference and result display. It defines routes for the home page, image upload, prediction and chatbot interaction. When an image is submitted, the application checks that a file exists, verifies the permitted file type, generates a safe filename if files are stored, and handles invalid input gracefully. The image is read, resized, converted to the required colour format, normalized and reshaped to the input shape of the saved model (an H5 file). The model returns an array of scores, the index with the highest score is mapped to a class name, and a result page is rendered. Raw stack traces should not be shown to users, and errors should be logged in a controlled way.

The frontend is a single web page with a title bar, a file chooser, a Predict button and a chatbot panel (Fig. 5.3). After prediction a result page shows the uploaded image beside descriptive text on the predicted class, together with a "Go To Market" button intended to give access to market-related information (Fig. 5.4).

G. Kannada Chatbot

The chatbot, named "Areca Assistant", is intended to reduce the language barrier between a technical prediction and the farmer's next question. The chat panel offers a language selector and a text box and is introduced as an arecanut farming assistant that can be asked about diseases, fertilizers and harvesting. A user may ask what the predicted class means, which symptoms are commonly associated with it, what to observe next and when to seek expert support. By design, responses should be short and clear in Kannada, should avoid unsupported claims, should state that image predictions are preliminary, and should direct the farmer to a qualified agricultural officer when symptoms are uncertain or severe. The

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

chatbot should also not prescribe pesticides, chemical dosages or treatment schedules without a verified agricultural source.

H. NLP-Based Query Processing

In general terms, a farmer's text query q is processed by a normalization or NLP function N and, optionally using the predicted disease class c as context, a response function R selects or generates the answer a that is displayed to the user, as in (6).

$$a = R(N(q), c) \tag{6}$$

Depending on the implementation, N may perform text normalization and keyword or intent matching, and R may select from prepared responses (rule-based), search a curated set of Kannada question–answer pairs (retrieval-based), use an NLP model to classify intent or extract entities, or generate text with a language model, which would require careful grounding and safeguards.

I. Farmer Interaction and Output Generation

Algorithm 3.1 summarizes the processing performed for each uploaded image. The output shown to the farmer consists of the predicted class, the uploaded image, descriptive text on symptoms, causes and management practices, and the chatbot panel for follow-up questions. The result is presented as a preliminary AI prediction for which expert confirmation is recommended.

Algorithm 3.1: Inference for one uploaded leaf image

1. Receive the uploaded file; reject missing or unsupported file types.
2. Read the image, convert to RGB, resize to 227 × 227 and rescale by 1/255 to obtain X' (1).
3. Add a batch dimension and compute the scores z and probabilities p with the saved model (2), (3).
4. Select the predicted index $\hat{y} = \arg\max p$ (4) and map it to a class label using the training class order.
5. Retrieve the symptom and management text stored for the predicted class.
6. Render the result page with the prediction and the chatbot panel, passing the class as optional context c (6).

IV. IMPLEMENTATION

A. Development Environment

The software environment listed in the project report is summarized in Table 4.1. Python is the main programming language; TensorFlow and Keras are used to build, train and load the CNN, OpenCV for image reading, resizing and colour conversion, NumPy for array operations, Matplotlib for accuracy and loss plots, and scikit-learn for evaluation metrics. The trained model is saved in H5 format and loaded by the Flask application. The version numbers in the report are minimum requirements and not necessarily the installed versions.

TABLE IV 1 SOFTWARE ENVIRONMENT LISTED IN THE PROJECT REPORT

Component	Details
Language	Python 3.8 or later
Deep learning	TensorFlow ≥ 2.3.1, Keras ≥ 2.4.3
Image processing / numerics	OpenCV ≥ 4.7.0.68, NumPy ≥ 1.18.5
Visualization / evaluation	Matplotlib ≥ 3.3.2, scikit-learn
Web application	Flask (model inference service)

B. Dataset Preparation Module

The dataset consists of labelled images grouped into training and validation folders, with each folder name corresponding to a class recognized by the model. Before training, images should be checked for incorrect labels, duplicates, unreadable files and severe quality problems, and the split should avoid placing images of the same plant or capture session in both training and validation data, which would inflate the measured accuracy. A separate test set is recommended for final performance reporting.

C. Model Training and Saving

During training, batches of labelled images are supplied to the network, the loss function measures the difference between predictions and true labels, and the Adam optimizer updates the parameters. Validation data are used to monitor generalization, and training and validation accuracy and loss are plotted to inspect learning behaviour and overfitting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

The configuration is given in Table 4.2; the training log shows 262 steps per epoch at a batch size of 32. The class-index mapping must be stored or reconstructed in the order used during training.

TABLE IV 2 TRAINING CONFIGURATION

Parameter	Value
Input size	227 × 227 × 3
Batch size / epochs	32 / 50
Optimizer	Adam
Training preprocessing	Rescale 1/255; rotation, zoom, horizontal flip
Validation preprocessing	Rescale 1/255 only

D. Web Application Module

The web application loads the saved model once and serves the home page, upload, prediction and chatbot routes described in Section 3.6. It performs the inference procedure of Algorithm 3.1 and renders the result page with the predicted class and the stored descriptive text. If upload history or user information is stored, only necessary information should be collected and access to it restricted.

E. Chatbot Module

The chatbot module is embedded in the page as a chat panel with a language selector. It receives the typed query, applies the query processing of Section 3.8 and returns a response that is displayed in Kannada script. In the captured example (Fig. 5.3) the user greets the assistant, asks about “yellow leaf disease” and receives replies in Kannada. The knowledge source and processing technique of this module remain to be stated by the authors.

F. Module Interaction

Table 4.3 summarizes how the modules interact: the dataset module feeds the training module, which produces the saved model used by the web application, while the chatbot module runs beside the web application and can receive the predicted class as context.

TABLE IV 3 MODULES, INPUTS, PROCESSING AND OUTPUTS

Module	Input	Processing	Output
Dataset preparation	Labelled images in class folders	Training/validation split; rescaling; augmentation (training only)	Batches of 227 × 227 × 3 images with labels
Model training	Image batches	Adam optimization for 50 epochs; validation monitoring	Saved H5 model; accuracy and loss history
Web application	Uploaded leaf image	Validation, preprocessing, inference, class mapping	Result page with class and management text
Chatbot	Typed query (and optional class)	Normalization and response selection	Kannada response in chat panel

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section reports the results that are documented in the project report: the training and validation behaviour of the CNN, a classification report and a confusion matrix, and the outputs of the web application and chatbot. Results that were not measured are identified as such and are not estimated.

A. Experimental Setup and Dataset Information

The experiments use the final saved model and the same preprocessing pipeline as the deployed application. The label set visible in the evaluation output covers banana, cotton and arecanut-related categories; arecanut-related labels include Mahali_Koleroga (Koleroga or fruit rot) and yellow leaf disease, alongside generic healthy-leaf, healthy-nut and healthy-trunk labels. The training log shows about 917–966 s per epoch (about 4 s per step) for the final two epochs.

B. Evaluation Metrics

Accuracy is the fraction of correctly classified samples. Per-class precision, recall and F1-score are computed from true positives (TP), false positives (FP) and false negatives (FN), as in (7)–(9), and summarized with macro (unweighted) and weighted averages. These measures are meaningful only when computed on data that were not used for parameter updates or model selection.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

$$\text{Precision} = TP / (TP + FP) \tag{7}$$

$$\text{Recall} = TP / (TP + FN) \tag{8}$$

$$F1 = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \tag{9}$$

C. Training and Validation Results

Table 5.1 gives the accuracy and loss at the final epoch (50/50). The training accuracy was 99.03% and the validation accuracy 99.67%; these are training-set and validation-set values, not test-set results. Validation accuracy slightly above training accuracy can occur when augmentation and regularization are active only during training.

TABLE V 1 FINAL-EPOCH TRAINING AND VALIDATION RESULTS (EPOCH 50 OF 50)

Metric	Training	Validation
Accuracy	99.03%	99.67%
Loss	0.0290	0.0095

Fig. 5.1 shows accuracy and loss over the 50 epochs. Training accuracy rises rapidly and approaches 99–100%, and training loss decreases steadily. Validation accuracy and loss fluctuate in the early and middle epochs, with several temporary drops and spikes, and then stabilize at high accuracy and low loss. This indicates convergence on the training and validation data; it does not by itself show performance on unseen images.

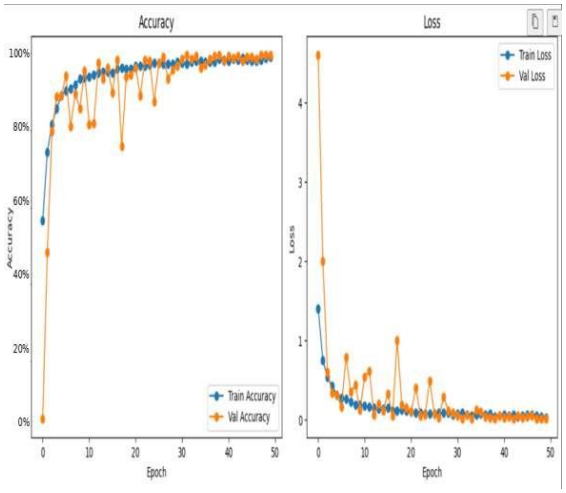


Fig. 5 1 Training and validation accuracy (left) and loss (right) over 50 epochs.

D. Classification Report

A scikit-learn classification report was produced for the trained model on 8,143 samples (Table 5.2). It reports an overall accuracy of 1.00 (rounded to two decimals), macro-averaged precision, recall and F1-score of 0.98, 0.99 and 0.98, and weighted averages of 1.00. Most classes score 1.00, including Mahali_Koleroga (367 samples). The lowest precision is for Banana Split Peel (0.63, 12 samples), and Banana Fruit-Scarring Beetle has the lowest recall among the rows shown (0.86). Rows for the remaining classes, including yellow leaf disease, are truncated in the captured output.

TABLE V 2 CLASSIFICATION-REPORT VALUES REPORTED FOR 8,143 EVALUATION SAMPLES (SELECTED ROWS)

Class / average	Precision	Recall	F1-score	Support
Mahali_Koleroga	1.00	1.00	1.00	367
Banana Fruit-Scarring Beetle	1.00	0.86	0.92	69
Banana Split Peel	0.63	1.00	0.77	12
Accuracy	–	–	1.00	8143
Macro average	0.98	0.99	0.98	8143
Weighted average	1.00	1.00	1.00	8143

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

E. Confusion Matrix

Fig. 5.2 shows the confusion matrix generated for the same label set, with actual classes on the vertical axis and predicted classes on the horizontal axis. Counts are concentrated on the diagonal, and in the legible portion of the figure the off-diagonal entries are small. The matrix also shows that the label set is multi-crop, so it should be read as an evaluation of the combined model and not as an arecanut-only result. Near-perfect scores on a random split can reflect visual similarity between images of the same plant or capture session, and classes with very few samples (for example 7 or 12 images) give unstable per-class estimates.

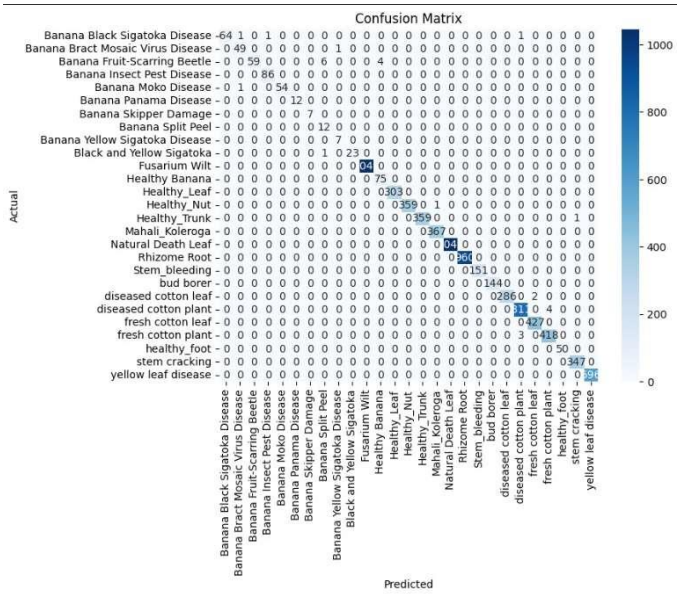


Fig. 5 2 Confusion matrix generated by the project team for the evaluation set (multi-crop label set; split to be confirmed).

F. Application Outputs and Chatbot Functionality

Fig. 5.3 shows the main page of the web application. The user selects a leaf image with the file chooser and presses Predict; the Areca Assistant panel on the right offers a language selector and a text box, and the captured exchange shows a greeting followed by a reply in Kannada script to the query “yellow leaf disease”. Fig. 5.4 shows the result page for one uploaded image, which displays the predicted class “Areca Yellow Leaf Disease”, the input image with yellowing and brown spots, and descriptive text on symptoms, causes and management, including site selection, drainage, balanced fertilization, removal of infected palms, vector control and quarantine. In the captured result page the descriptive text is in English. The management text includes specific fertilizer and insecticide details that should be verified by an agricultural expert, since the system is not intended to prescribe chemical treatments without a verified source. These screenshots show functionality for single examples and are not evidence of accuracy or of chatbot quality.



Fig. 5 3 Main page of the web application with the Kannada-capable “Areca Assistant” chatbot panel.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152



Fig. 5 4 Result page for an uploaded image, showing the predicted class and management information.

G. Testing and Evaluation Status

Table 5.3 summarizes what has been demonstrated and what has not. The report documents functional behaviour through screenshots of single examples, but it does not contain systematic test cases, response-time measurements or user studies, and these are therefore reported as not evaluated.

TABLE V 3 EVALUATION STATUS OF THE PROPOSED SYSTEM

Aspect	Evidence in the report	Status
Training/validation accuracy and loss	Epoch-50 values and curves (Table 5.1, Fig. 5.1)	Reported
Aggregate precision, recall, F1	Classification report, 8,143 samples (Table 5.2)	Reported; split to be confirmed
Confusion matrix	Fig. 5.2	Reported; split and scope to be confirmed
Independent held-out test set	Recommended in the report	Not reported
Image upload and prediction display	Figs. 5.3 and 5.4	Demonstrated on single examples
Kannada chatbot reply	One exchange in Fig. 5.3	Demonstrated on one example; no quantitative evaluation
Inference / response time	–	Not measured
User study or field trial with farmers	–	Not conducted

H. Comparison with a Manual Workflow

Table 5.4 compares the proposed workflow with manual inspection qualitatively. It describes what the system automates and what still depends on the farmer or an expert; it is not a measured claim that the system outperforms existing tools.

TABLE V 4 QUALITATIVE COMPARISON OF MANUAL INSPECTION AND THE PROPOSED SYSTEM

Aspect	Manual inspection / general search	Proposed system
Image assessment	Depends on the observer's experience	Model predicts among trained classes
Time to initial screening	Varies with field conditions and access to advice	Includes upload, inference and page response; actual time not measured
Language accessibility	Depends on available information and expert language	Kannada chatbot intended to improve access
Follow-up questions	May require contacting another person or searching separately	Can be asked through the integrated chatbot
Reliability	Depends on experience and evidence available	Depends on dataset, model, preprocessing and class coverage
Expert confirmation	Needed for uncertain or severe cases	Still recommended; decision-support aid

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511152

I. Discussion and Limitations

The results show that the AlexNet-style CNN fits the training and validation data well and that the pipeline from upload to prediction and chatbot support works end to end. The practical contribution is the combination of a leaf-image classifier and a Kannada chatbot in one user workflow, which may reduce the gap between a technical prediction and a farmer's understanding of what it means. Several factors limit what can be concluded. The label set is multi-crop whereas the application targets arecanut; the evaluation split is unspecified; arecanut classes described in the report, such as Anaberoga and leaf spot, do not appear in the captured evaluation output; no independent test set or field trial is reported; and the chatbot has not been evaluated for answer correctness, Kannada quality or usability. Novelty and effectiveness are therefore claimed only to the extent supported by the implementation.

VI. CONCLUSION

This paper presented a preliminary system that combines an AlexNet-style sequential CNN for plant-image classification with a Kannada chatbot interface in a Flask web application for arecanut farmers. After 50 epochs the model reached 99.03% training and 99.67% validation accuracy, and a classification report on 8,143 samples gave macro-averaged precision, recall and F1-score of 0.98, 0.99 and 0.98. The web application returns the predicted class with symptom and management information, and the Areca Assistant panel allows follow-up questions to be asked in Kannada.

For farmers, the system offers a quick first screening of a leaf image and information in a familiar language, which may help them seek expert advice sooner, avoid acting on unverified informal advice and become familiar with AI-assisted crop monitoring; these benefits require field validation. The contributions of the work are the deployment of a deep-learning classifier in a web workflow, the design of a Kannada chatbot interface for farmer questions, the combination of both in one user experience, and the identification of evaluation measures, limitations and responsible-use requirements for field deployment.

The current limitations are the multi-crop label set, the unconfirmed evaluation split, the absence of an independent test set and field trials, the limited resolution and variety of the documented images, and the lack of a quantitative evaluation of the chatbot. The system should be treated as a decision-support aid, with expert confirmation recommended for uncertain or serious crop-health problems.

VII. FUTURE SCOPE

The limitations identified above point to the following improvements.

A. Data and Model

The arecanut dataset should be expanded with images from different regions, seasons, devices and disease stages, with expert-verified labels, and a separate, representative test set should be added so that class-wise results can be published. Generalization can be improved with data-quality checks, class balancing and suitable augmentation, and lightweight models or optimized inference should be explored for mobile and low-resource devices.

B. Chatbot and Knowledge Sources

Kannada NLP coverage can be improved with reviewed question–answer pairs and agricultural terminology, and the chatbot can be connected to a verified agricultural knowledge base maintained by qualified experts. Confidence-aware responses with a clear escalation path to an expert should be provided when predictions are uncertain. Voice-based farmer interaction, building on existing Kannada speech recognition work [8], [12], would suit users with limited literacy, but it must be implemented and tested before it is claimed.

C. Deployment and Field Validation

A mobile application integration can make the tool easier to use in the field. Usability studies with Kannada-speaking farmers and extension workers, and evaluation of field outcomes such as the time to obtain advice, user comprehension and whether the system supports better decisions, are required. After validation of the arecanut version, additional crops or regional languages can be supported, and the application should be deployed with suitable privacy, security, monitoring and maintenance procedures.

ACKNOWLEDGMENT

The authors express their sincere gratitude to their guide, Dr. Deepa V B, Assistant Professor, Department of Information Science and Engineering, for continuous encouragement and invaluable guidance throughout the project work. They



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511152

thank Mrs. Vishrutha Rani S. N. for her unwavering support and encouragement, and Dr. Raghavendra R J, Associate Professor and Head, Department of Information Science and Engineering, and Dr. Y Vijaya Kumar, Principal, Jawaharlal Nehru New College of Engineering, Shivamogga, for their immense support and motivation. The authors are grateful to the Department of Information Science and Engineering and the institution for the knowledge and resources provided, and to the teaching and non-teaching staff of the department and their families for their constant support and encouragement.

REFERENCES

[1]. Mamatha Balipa, Pallavi Shetty, Arhath Kumar, Puneeth B R and Adithya, "Arecanut Disease Detection Using CNN and SVM Algorithms", in *Proc. International Conference on Artificial Intelligence and Data Engineering (AIDE)*, 2022.
[2]. Madhu B G, Ram Kumar G, Shreehari S Rao, Chandrashekara M and Amith Shetty A R, "Detection of Diseases in Arecanut Using Convolutional Neural Network", in *Proc. Second International Conference on Advances in Information Technology (ICAIT-2024)*, 2024.
[3]. Meghana D R and Prabhudeva, "Image Processing based Arecanut Diseases Detection Using CNN Model", *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*.
[4]. Veeresha R K, Lathish Kumar N D, Samarth S Shetty, Shrajan G Prasad, Swaroop S Poojary, Shilpa Karegoudra and Hasan Koten, "Deep Learning-Based Classification of Areca Nut Yellow Leaf Disease with ResNet-50 CNN", in *Proc. International Conference on Recent Advances in Science & Engineering Technology (ICRASET)*, 2024.
[5]. Beena K, Sangeetha V, Hrishikesh Keraba Lohar, Nidhi G D, Varshini S and Yogesh K N, "Arecanut Disease Prediction using MobileNet V2, ResNet and VGG-16 Framework", in *Proc. Fourth International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS-2024)*, 2024.
[6]. G. Amulya and R. Saranya Lakshmi, "AI-Based Crop Disease Detection and Multilingual Farmer Assistance System", *International Journal on Science and Technology (IJSAT)*, 2026.
[7]. Akash Palepu, Pasam Sai Rakesh and Sonia Jenifer Rayen, "Agriguide: AI Powered Voice Assistant for Farmers", *International Journal for Multidisciplinary Research (IJFMR)*, 2025.
[8]. Anjani, Pooja, Ranjitha V P and Manjunath Acharya, "Kannada Speech Recognition Enquiry System for Farmers", *International Journal of Engineering Research & Technology (IJERT)*, 2019.
[9]. Shuhan Lei, Jianbiao Luo, Xiaojun Tao and Zixuan Qiu, "Remote Sensing Detecting of Yellow Leaf Disease of Arecanut Based on UAV Multisource Sensors", MDPI, 2021.
[10]. Ashritha K P, Navaneeth Bhaskar, Lakshmi H N, Shaik Jaffar Hussain, Akepati Sankar Reddy and B Rupa Devi, "Deep Learning-Based Arecanut Disease Detection and Smart Cluster Management with Automated Pesticide Spraying System", in *Proc. Second International Conference on Networks and Soft Computing (ICNSoC)*, 2025.
[11]. Amit Gupta, Rajeshwari B S, Ambadas B, Shreyas J and Bhuvan G, "Analysis of Areca Nut Leaf Pathology and Recommendation System Using Generative AI", in *Proc. Second International Conference on Networks, Multimedia and Information Technology (NMITCON)*, 2024.
[12]. Chandrika Prasad, Veena Gode Swamy Rao, J. Geetha and R China Appala Naidu, "An ASR Transformer-Based Model for Kannada Speech-to-Text Transcription", *Journal of Artificial Intelligence and Technology*, 2026.
[13]. Gunti Spandan and Prasannavenkatesan Theerthagiri, "Neural Machine Translation Model Using GRU with Hybrid Attention Mechanism for English to Kannada Language", GITAM School of Technology, GITAM University Bengaluru, India, 2024.
[14]. Abu Rayhan, Robert Kinzler and Rajan Rayhan, "Natural language processing: transforming how machines understand human language", 2020.
[15]. Charlie Luca, "Natural Language Processing (NLP) for Document Analysis", 2025.
[16]. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1097–1105, 2012.