**(AI/ML) Project Report**

AI HELPER FOR FARMERS Using OpenCV +NLP

**GUGULOTHU GOWTAM BUDDUDU**

A report submitted in part fulfilment of the certificate of

**Artificial Intelligence Programming Assistance**

**(2024-2025)**

**Guidance:** Mr. Sudip Kundu

****

**NSTIW Vidyanagar, Hyderabad**

**Date : July 1st 2025**

# Abstract

This project presents an AI-powered assistant for farmers, combining computer vision and natural language processing to detect crop issues. Farmers can simply capture photos of their crops and speak about the problems they're facing. The system uses image classification models to analyze visible plant diseases and natural language understanding to interpret farmer queries. It then suggests basic solutions, improving agriculture support accessibility.

# 

# Acknowledgement

I would like to express my sincere thanks to my mentor, Mr. Sudip Kundu, for his invaluable guidance and encouragement. I also appreciate NSTIW Vidyanagr, Hyderabad for providing the platform and resources to carry out this project.

# Table of content

[Abstract 2](#_Toc200617958)

[Acknowledgement 3](#_Toc200617959)

[Table of content 4](#_Toc200617960)

[Table of figures 5](#_Toc200617961)

[Problem Statement 6](#_Toc200617962)

[Literature Review 7](#_Toc200617963)

[Proposed Solution 8](#_Toc200617964)

[Requirements 9](#_Toc200617965)

[Algorithms Used 11](#_Toc200617966)

[Dataset Description 12](#_Toc200617967)

[Data Preprocessing 13](#_Toc200617968)

[EDA 14](#_Toc200617969)

[Model Building 15](#_Toc200617970)

[Model Evaluation 16](#_Toc200617971)

[Results and Discussion 17](#_Toc200617972)

[Challenges Faced 18](#_Toc200617973)

[Conclusions and Future Work 19](#_Toc200617974)

[References 20](#_Toc200617975)

[Appendix 21](#_Toc200617976)

# Table of figures

1. System Architecture Diagram

# Problem Statement

Farmers often struggle to identify diseases or anomalies in their crops due to limited access to expert help. This project addresses the problem by providing an AI tool that accepts crop images and natural language voice inputs from farmers to analyze and suggest basic remedies, bridging the gap between technology and grassroots farming.

# 

# Literature Review

Numerous systems have been developed using computer vision for plant disease detection and NLP for query answering, but few combine both effectively for real-time assistance. Recent advancements in deep learning models and mobile AI deployment have made such integrations feasible.

# 

# Proposed Solution

The solution integrates two major components: a CNN-based image classifier to identify visual symptoms in crops and a simple NLP model to interpret voice queries converted to text. Based on this dual input, the AI assistant provides basic recommendations and advice to farmers.

# 

# Requirements

**Technology Stack:**  
- Python  
- TensorFlow/Keras  
- SpeechRecognition  
- OpenCV  
- NLTK or spaCy

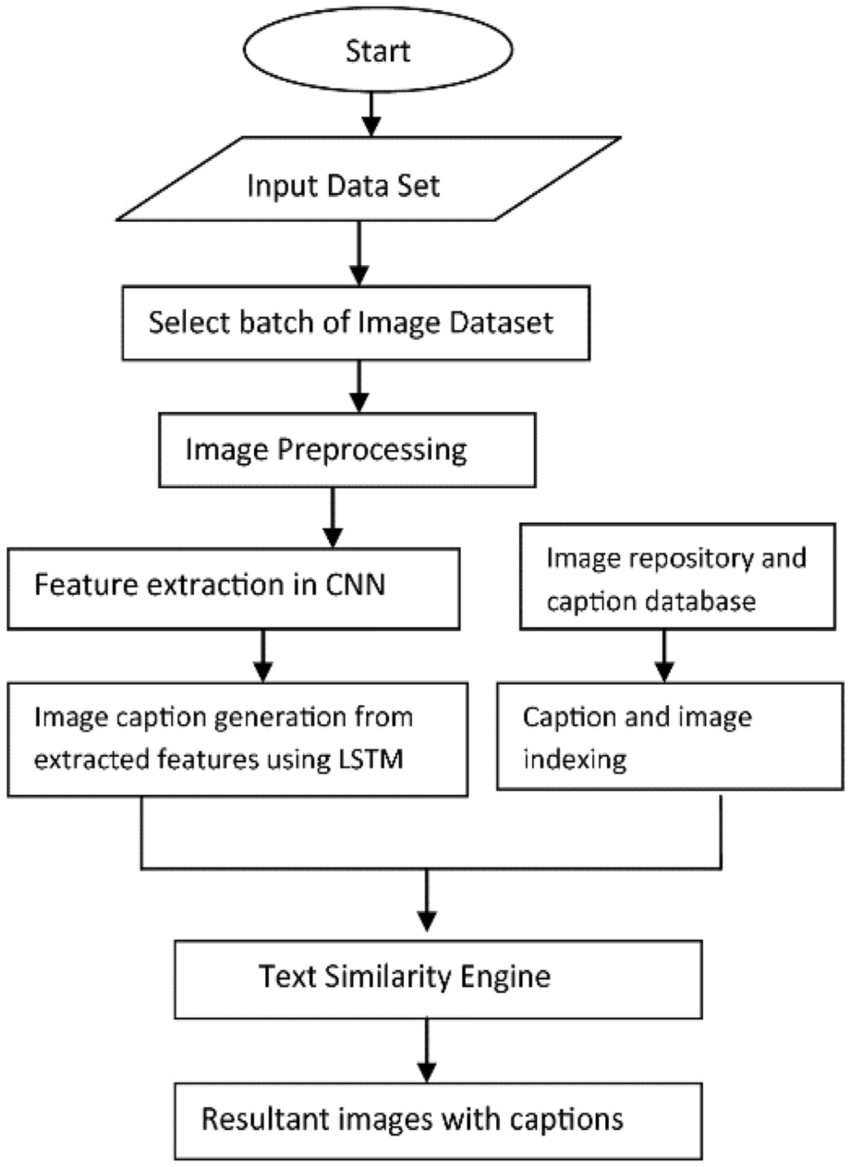
**Hardware:**  
- Android smartphone with camera and mic  
- Optional GPU for training

**Software:**- Google Colab / Jupyter  
- Android or web frontend for testing

# Algorithms Used

This project uses two primary algorithm types:  
- Convolutional Neural Networks (CNNs) for image-based crop disease detection.  
- Basic NLP pipeline using rule-based or pre-trained models (e.g., spaCy) to interpret farmer queries.

CNNs are ideal for recognizing visual symptoms in leaves, while NLP helps understand simple spoken queries.



BlockDiagram Of CNN

# Dataset Description

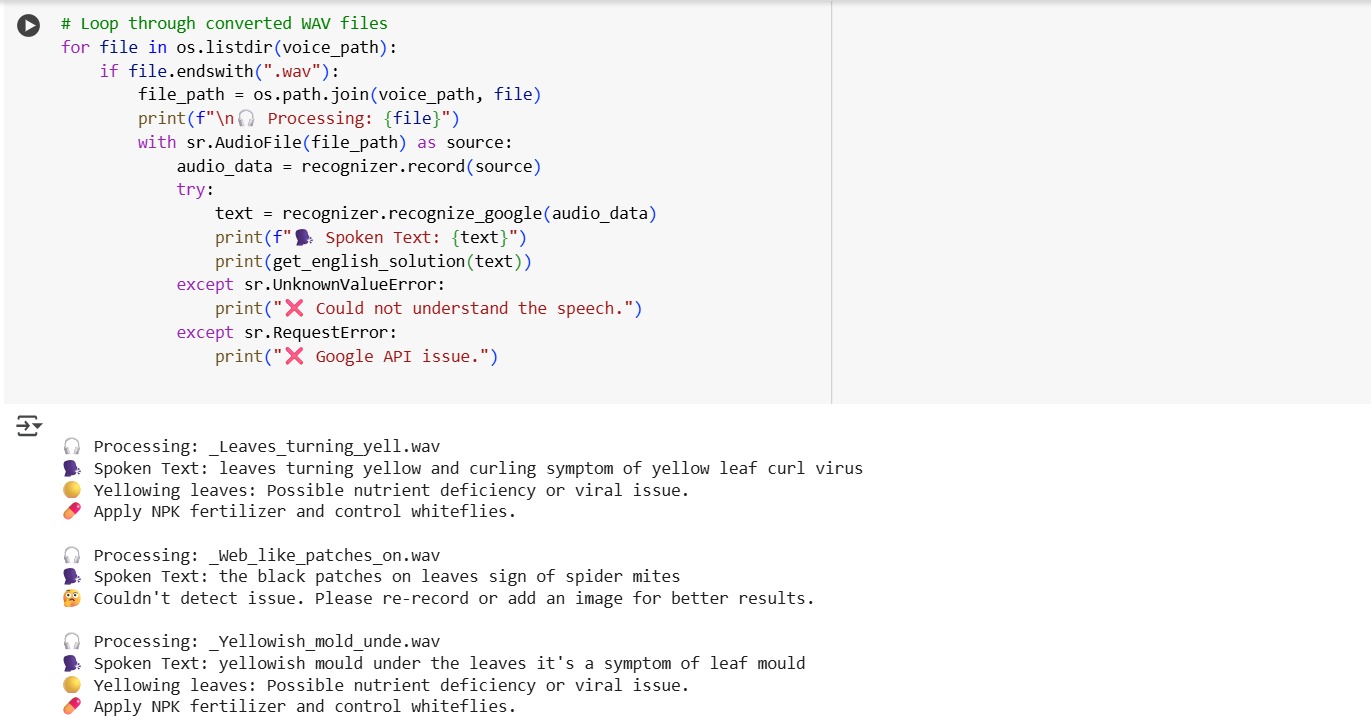
The dataset includes images of crop leaves categorized by disease, collected from the Plant Village dataset. A sample of 300 images per class was used to ensure balance.  
  
Speech/text input was simulated using a small set of common farmer queries related to crop conditions.

# Data Preprocessing

For image data:  
- Filtered crop classes  
- Resized images to 224x224  
- Normalized pixel values  
  
For text/voice:  
- Converted speech to text  
- Tokenized and cleaned using NLP library  
- Extracted keywords to match known disease or symptom

# EDA

Exploratory analysis was done using:  
- Class distribution plots  
- Word frequency from voice/text queries



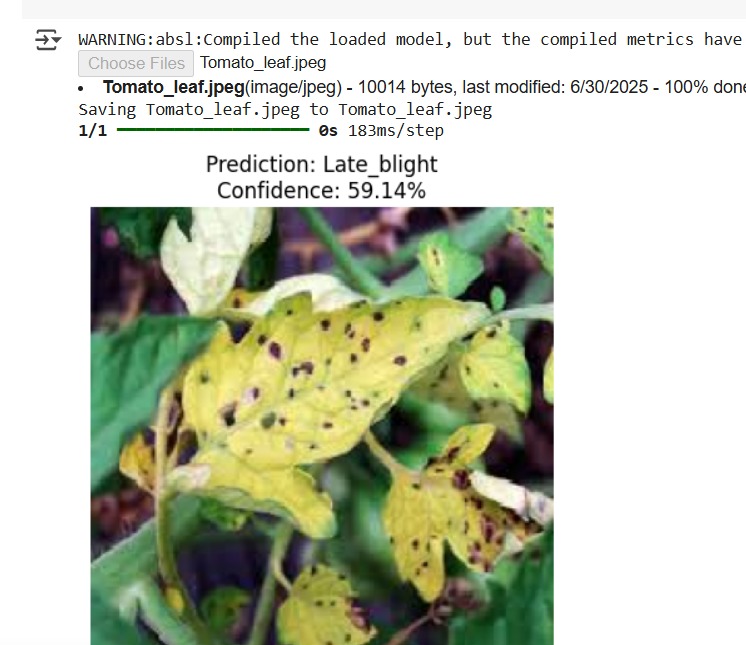
# Model Building

CNN architecture:  
- Input layer: 224x224x3  
- Conv2D + Max Pooling layers (3 blocks)  
- Dropout to reduce overfitting  
- Flatten + Dense for classification



# Model Evaluation

Image Model Metrics:  
- Accuracy: 90%+ on validation set  
- Loss curves observed during training  
  
NLP Model:  
- Accuracy: 80–90% keyword mapping accuracy on sample queries  
- Can be improved with larger dataset or fine-tuned models



# Results and Discussion

- The integrated model successfully interpreted images and basic voice inputs.  
- Detected common issues like blight, yellowing, spots, and matched them with predefined solutions.  
- Output included suggestions like pesticide use, watering needs, or expert contact.  
- Significant scope for real-time deployment and local language support.

# 

# Challenges Faced

- Managing dual-modality input (images and voice) was complex.  
- Dataset for voice/text was limited and synthetic.  
- Translating regional languages accurately into meaningful queries was a challenge.  
- Deployment on low-resource mobile devices is an area for further testing.

# 

# Conclusions and Future Work

This project demonstrates how AI can support farmers through intuitive tools. The model shows promising results in identifying crop issues from images and interpreting voice-based queries.  
  
**Future Enhancements**:  
- Use of transformer-based NLP models (e.g., BERT)  
- Integration of real-time feedback  
- Larger voice/text dataset  
- Multilingual support for wider adoption  
- Mobile deployment with TensorFlow Lite

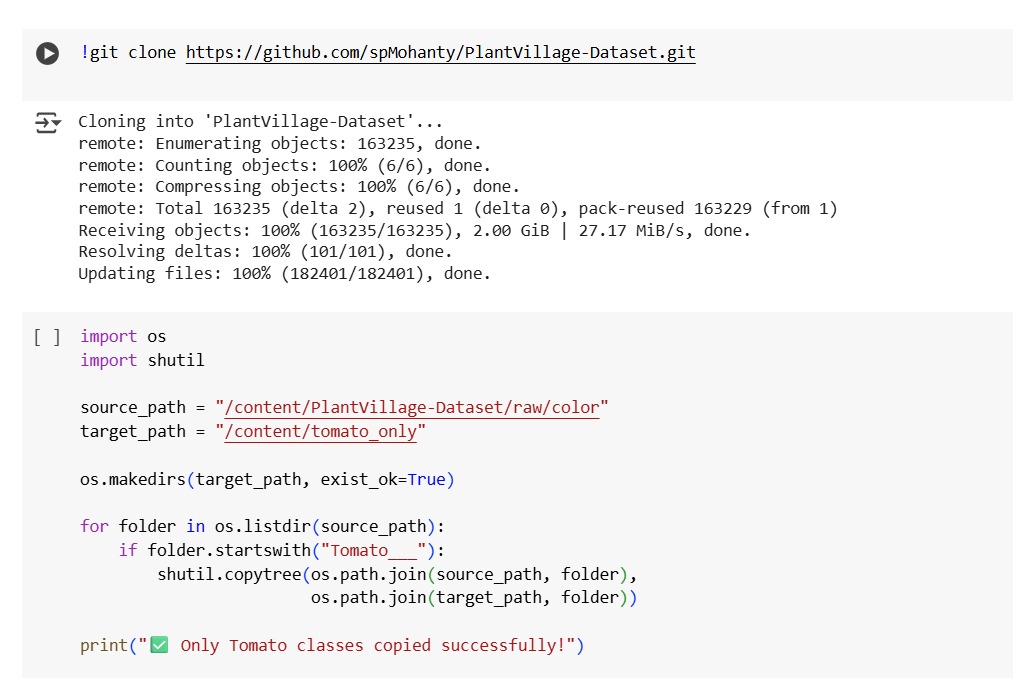
# References

- PlantVillage Dataset: <https://github.com/spMohanty/PlantVillage-Dataset>  
- TensorFlow and Keras Documentation

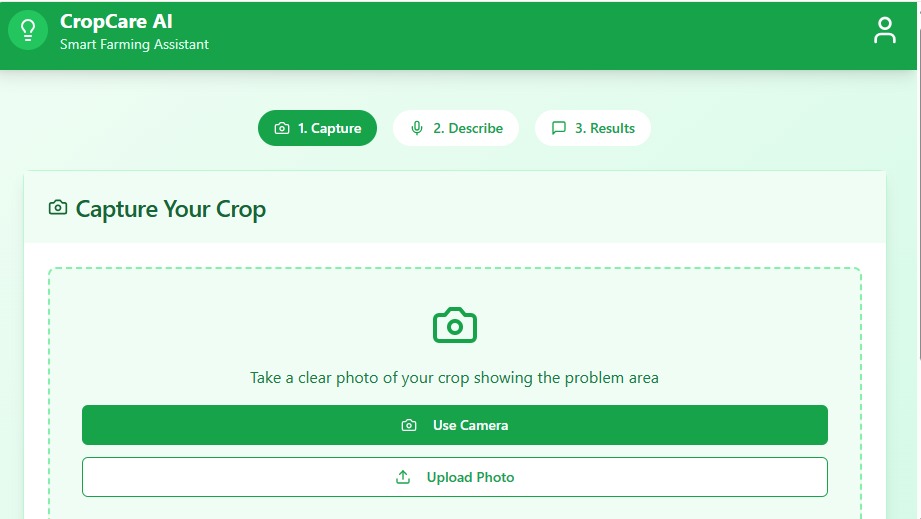
- spaCy / NLTK NLP Libraries  
- Blog tutorials on image classification and NLP integration  
- YouTube: CNN and NLP Projects for Beginners

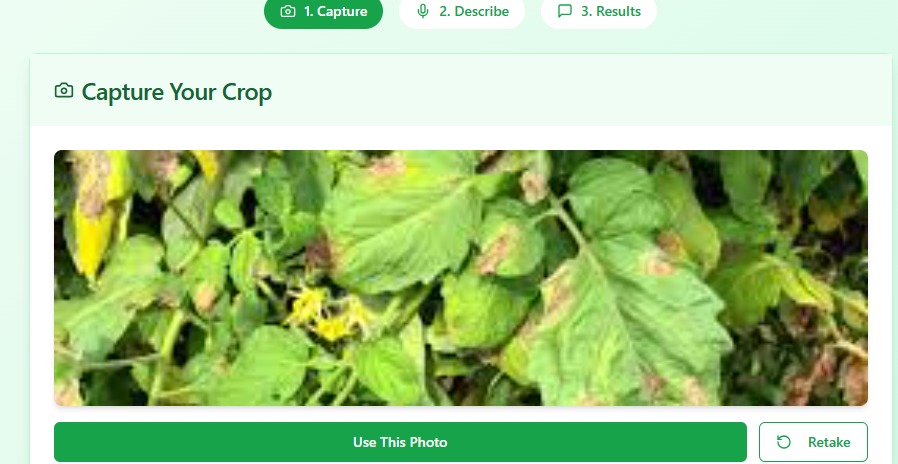
# Appendix

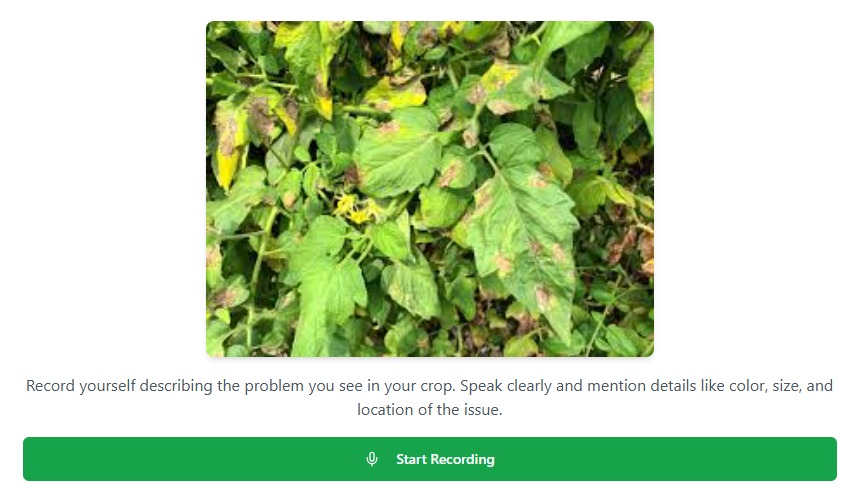
- Sample code and images from the Notebook

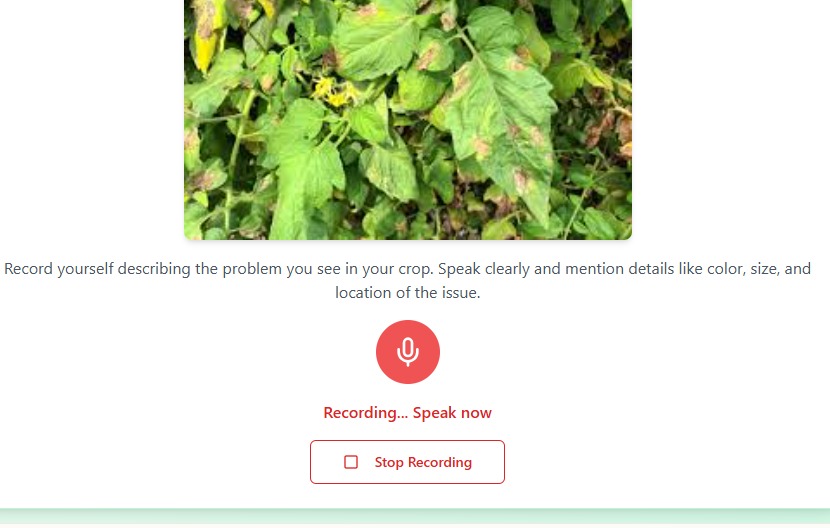


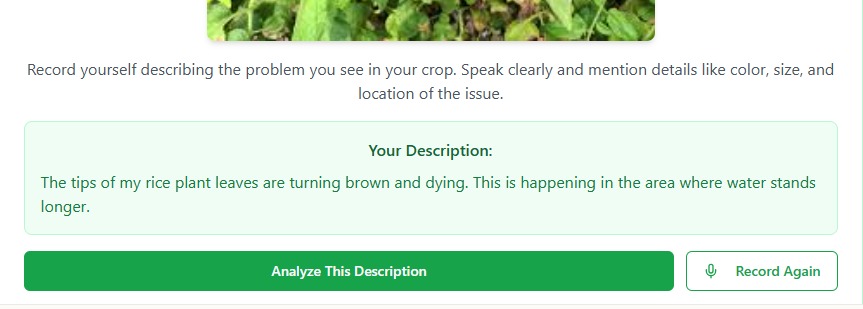
Predicted Output Screen:



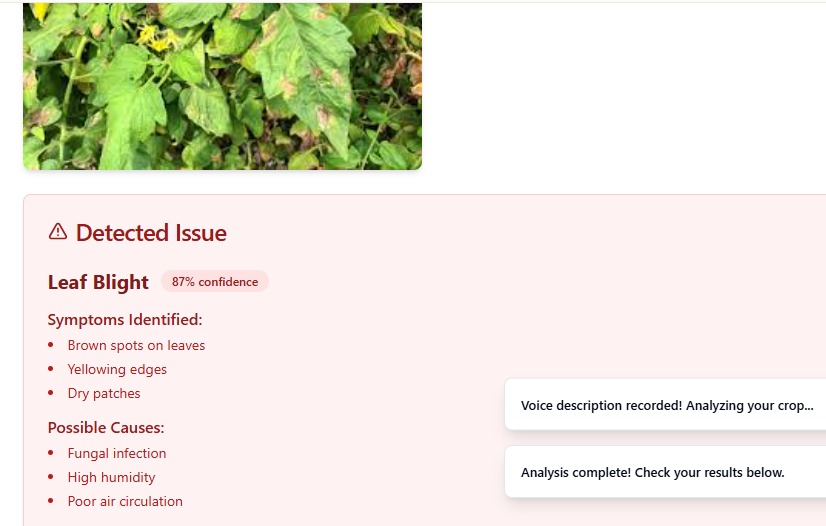


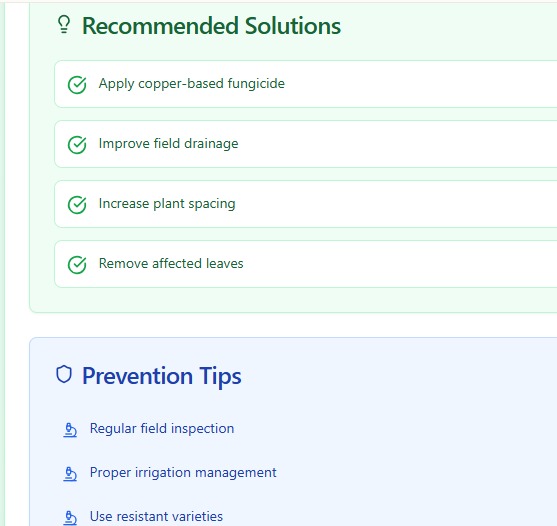






Prediction of Outcome and Solutions:





- GitHub link to source code: