# CROP DISEASE DETECTION

GROUP SEVEN

# Problem Statement

Crop diseases are a leading cause of reduced agricultural productivity worldwide, especially in regions where access to expert diagnosis is limited.

Farmers often struggle to identify diseases early, resulting in significant yield loss and increased use of harmful chemicals.

This project proposes a deep learning solution using Convolutional Neural Networks (CNNs) to detect and classify crop diseases from leaf images. The goal is to provide an accessible, image-based diagnostic tool that empowers farmers with timely, accurate disease identification.

# Our Project Objectives

01 Build a CNN model to classify crop diseases from leaf images.

02 Develop NLP methods to analyze farmers' textual symptom descriptions.

03 Integrate CNN and NLP outputs into a single multimodal diagnostic model.

04 Compare multimodal performance to image-only and text-only models.

05 Create a user-friendly API for farmers to upload photos and symptoms for instant diagnosis.

## What Do You Know About our CNN Data.

- Our data was from: https://www.kaggle.com/datasets/emmarex/plantdisease.
- The data contained 15 folders which contained our images classes.
- It had 3 crops:
  - Bell peppers – had 2 classes
  - Potato – had 3 classes
  - Tomato – had 10 classes

## What Do You Know About our NLP Data.

- We had difficulties find a data set that described symptoms of diseased crop so we synthesized our own data using prefix words, adjectives, possible environmental friendlypesticides and descriptions of the actual crop disease we could find from the internet.
- The end result was a data frame containing 1750 rows and 5 columns (description, crop, status of the crop, name of the disease and recommended pesticide)

# Data Cleaning & Preprocessing

01 Since our data was mostly images no cleaning was required for the CNN model

02 .For the NLP model we had to remove duplicates. Created new features for target class containing the original 15 classes in our CNN

03 Removed stop words, special characters and numbers, tokenized and lemmatized words to their parent word

04 Vectorized our clean text using TF-IDF then label encoded our target feature.

# CNN Architecture

**01** Image resizing: (128 × 128)
Input shape: (128, 128, 3)

**02** Data augmentation:

- Random flip
- Rotation
- Zoom

**03** Normalization:
Pixel values scaled to [0, 1]

**04** Layers:

- Conv2D ⟶ ReLU ⟶ MaxPooling
- Conv2D ⟶ ReLU ⟶ MaxPooling
- Conv2D ⟶ ReLU ⟶ MaxPooling
- Flatten
- Dense ⟶ ReLU
- Dropout
- Dense (softmax output)

**05**

- Loss: categorical crossentropy
- Optimizer: Adam

Model: "sequential_3"

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_3 (Conv2D) | (None, 126, 126, 32) | 896 |
| max_pooling2d_3 (MaxPooling2D) | (None, 63, 63, 32) | 0 |
| conv2d_4 (Conv2D) | (None, 61, 61, 64) | 18,496 |
| max_pooling2d_4 (MaxPooling2D) | (None, 30, 30, 64) | 0 |
| conv2d_5 (Conv2D) | (None, 28, 28, 128) | 73,856 |
| max_pooling2d_5 (MaxPooling2D) | (None, 14, 14, 128) | 0 |
| flatten_1 (Flatten) | (None, 25088) | 0 |
| dense_2 (Dense) | (None, 128) | 3,211,392 |
| dropout_1 (Dropout) | (None, 128) | 0 |
| dense_3 (Dense) | (None, 15) | 1,935 |

Total params: 3,306,575 (12.61 MB)
Trainable params: 3,306,575 (12.61 MB)
Non-trainable params: 0 (0.00 B)

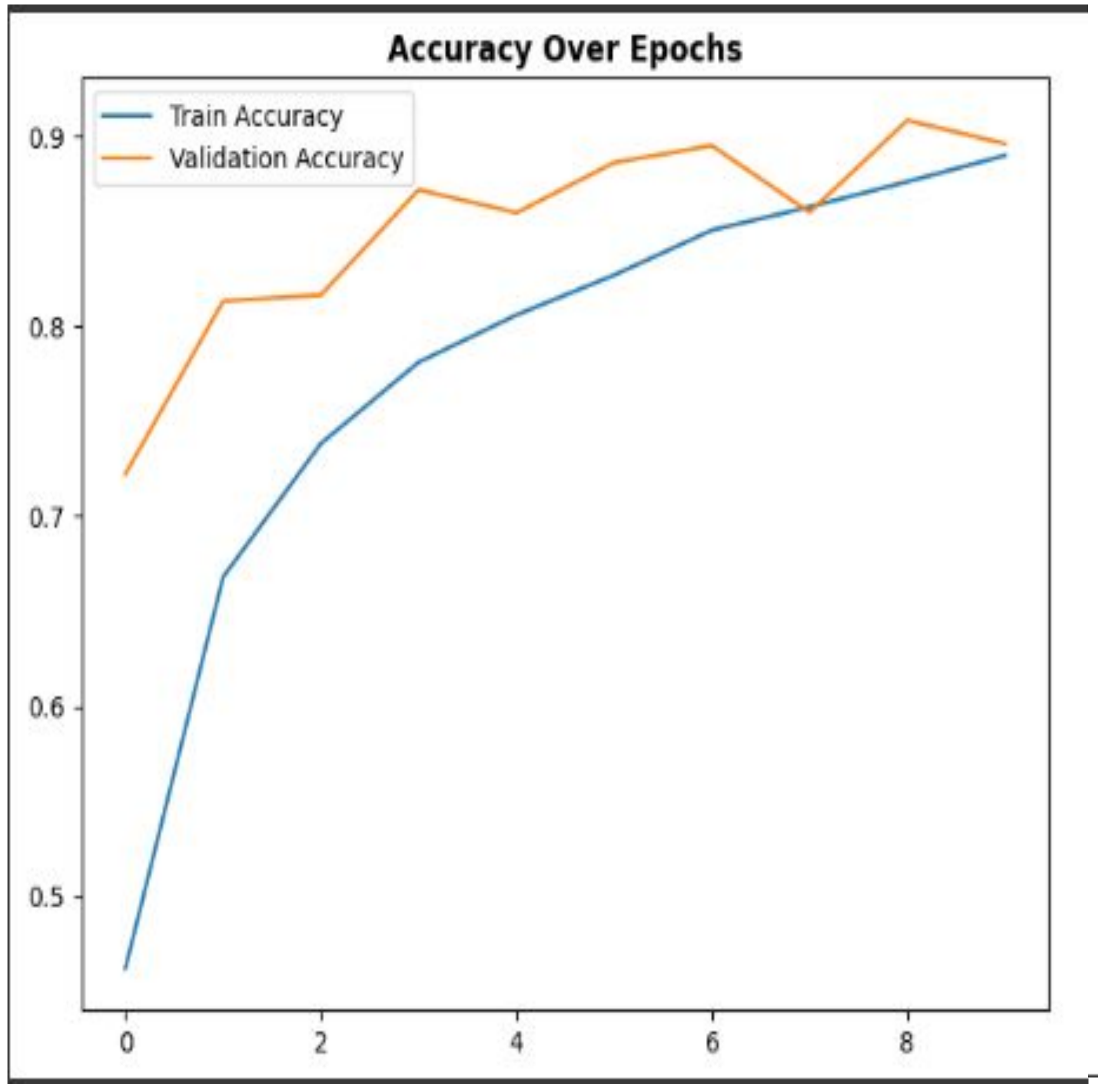
Ginyard International Co.

Home Service About Us Contact

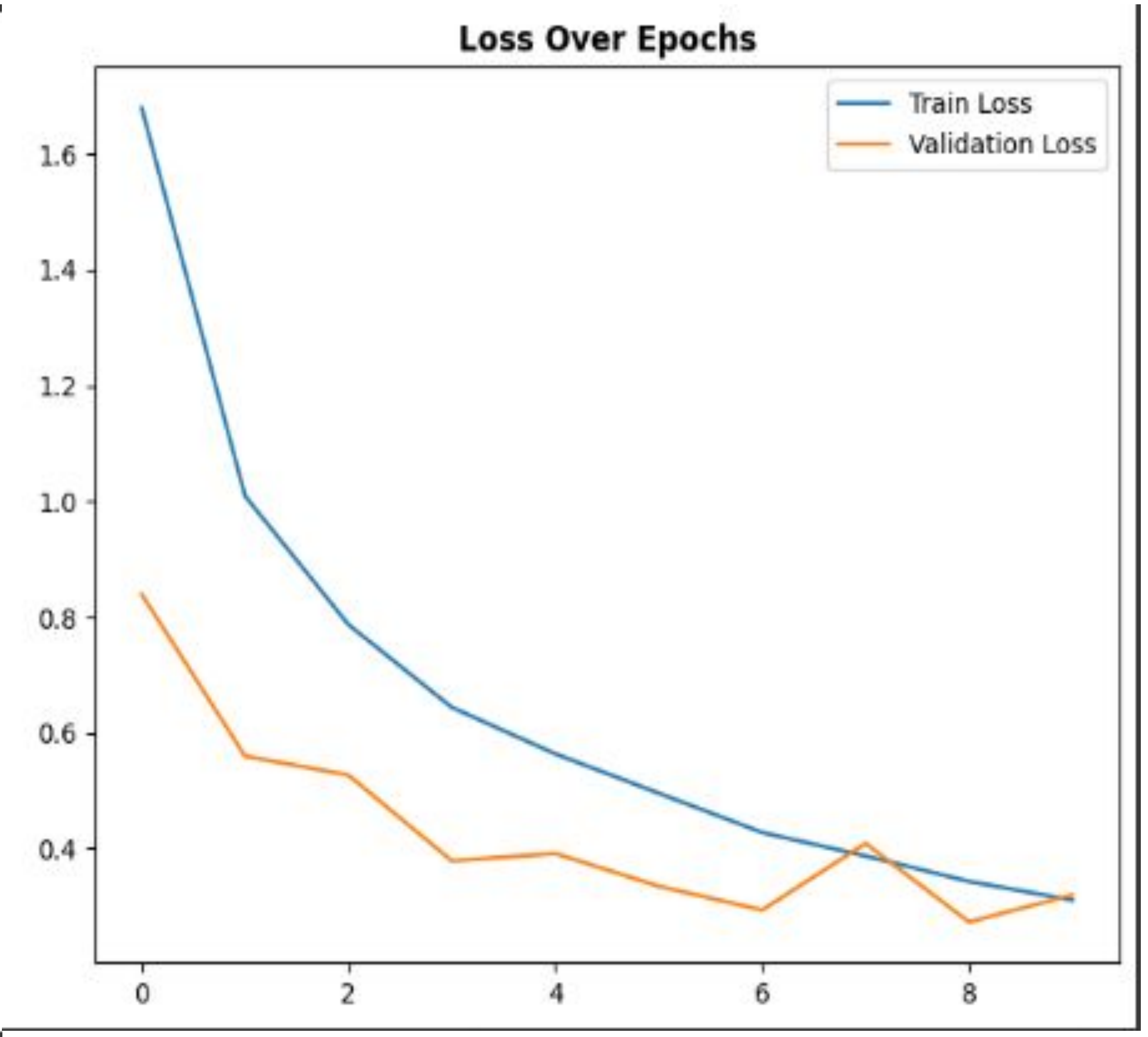
# CNN Performance

**Accuracy:** 0.8962
**Loss:** 0.3202

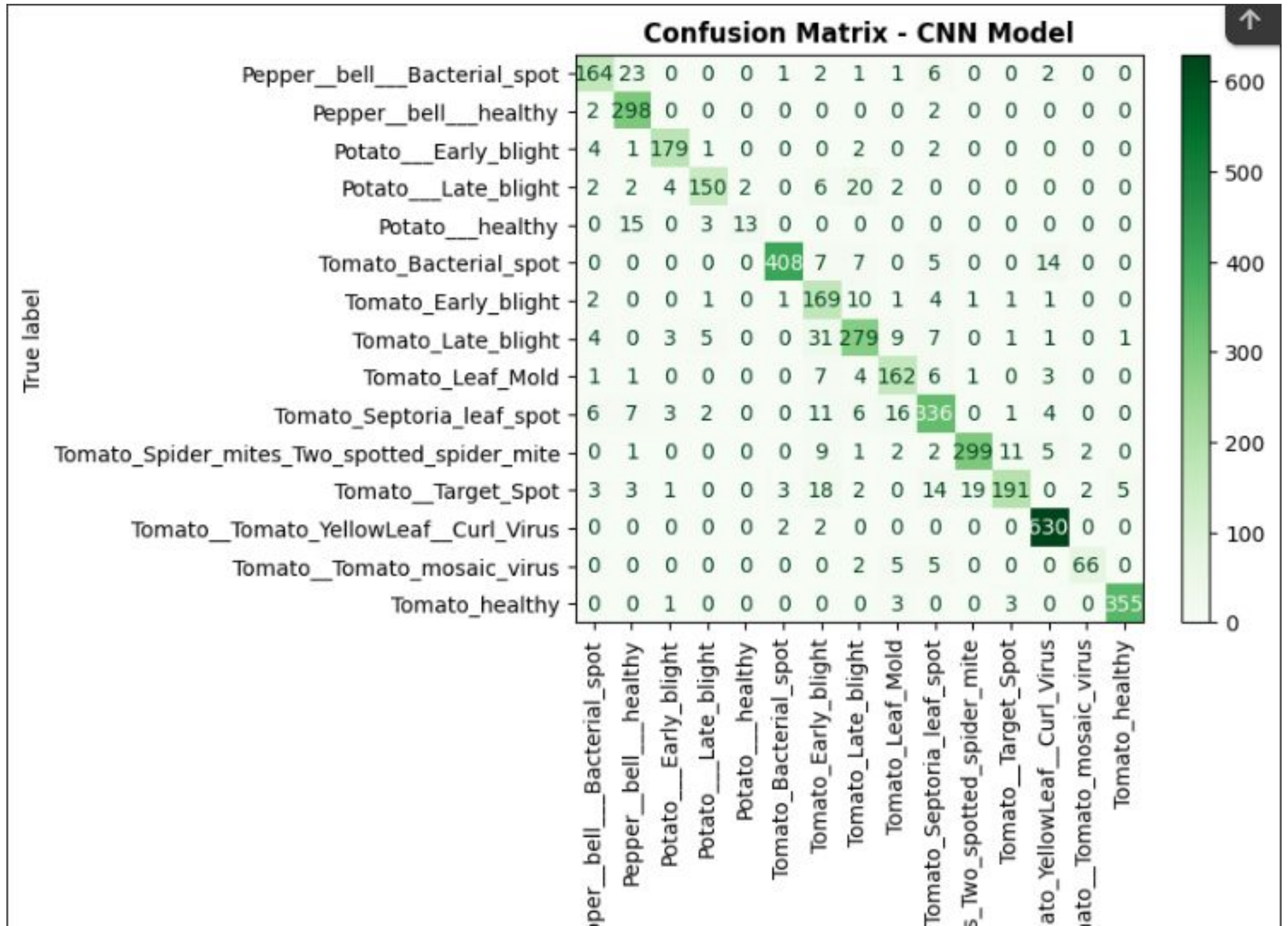
## Accuracy Over Epochs




## Loss Over Epochs

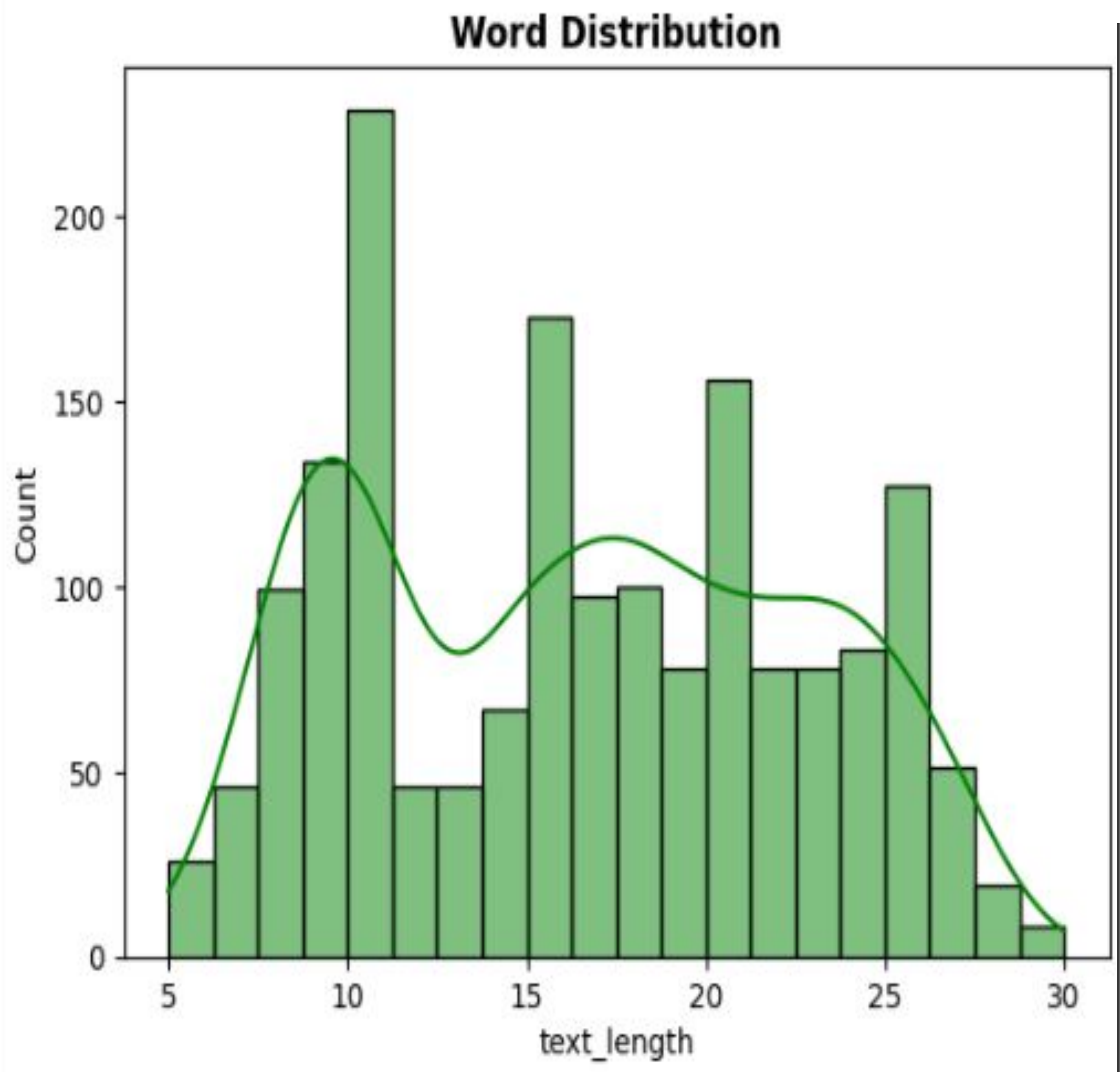



## Confusion matrix

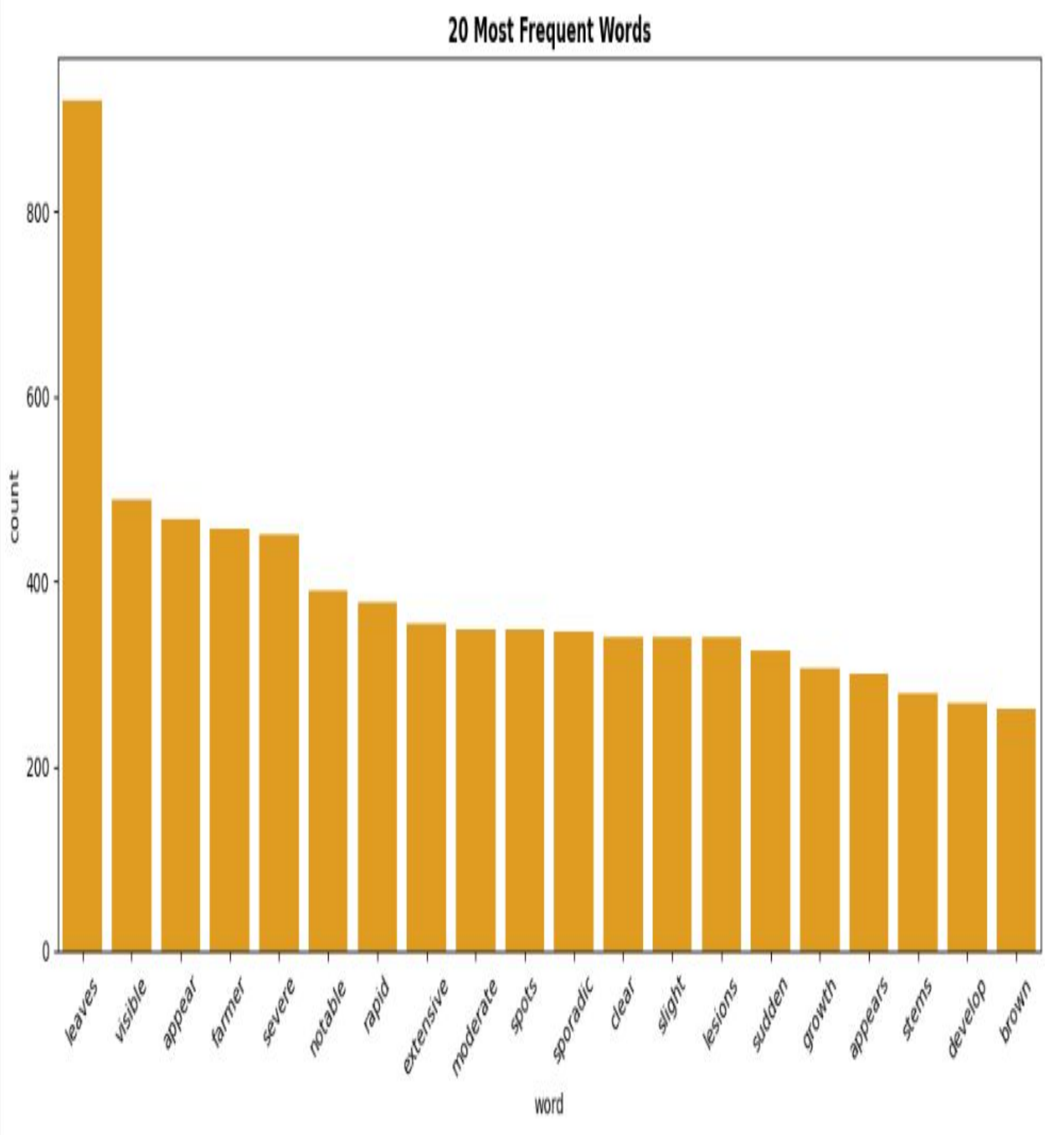


# NLP Data Pipeline

- Text pre-processing:
  - Lowercasing
  - Removing punctuation
- Vectorization:
  - CountVectorizer or TF-IDF
- Model:
  - Multinomial Naive Bayes
- Outputs:
  - Crop type
  - Disease/health label
  - Recommended pesticide




Word Distribution



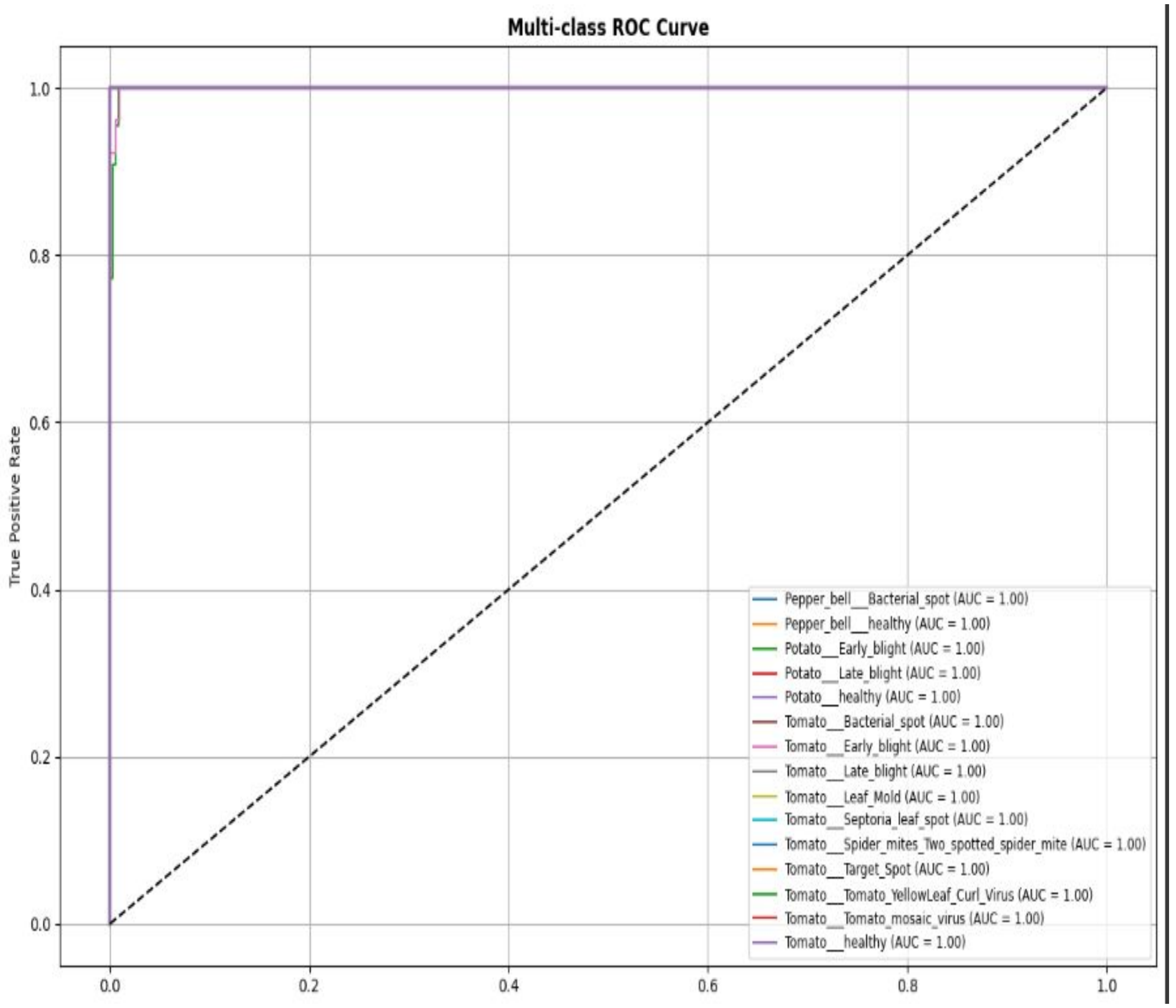

20 Most Frequent Words

Ginyard International Co.

Home Service About Us Contact

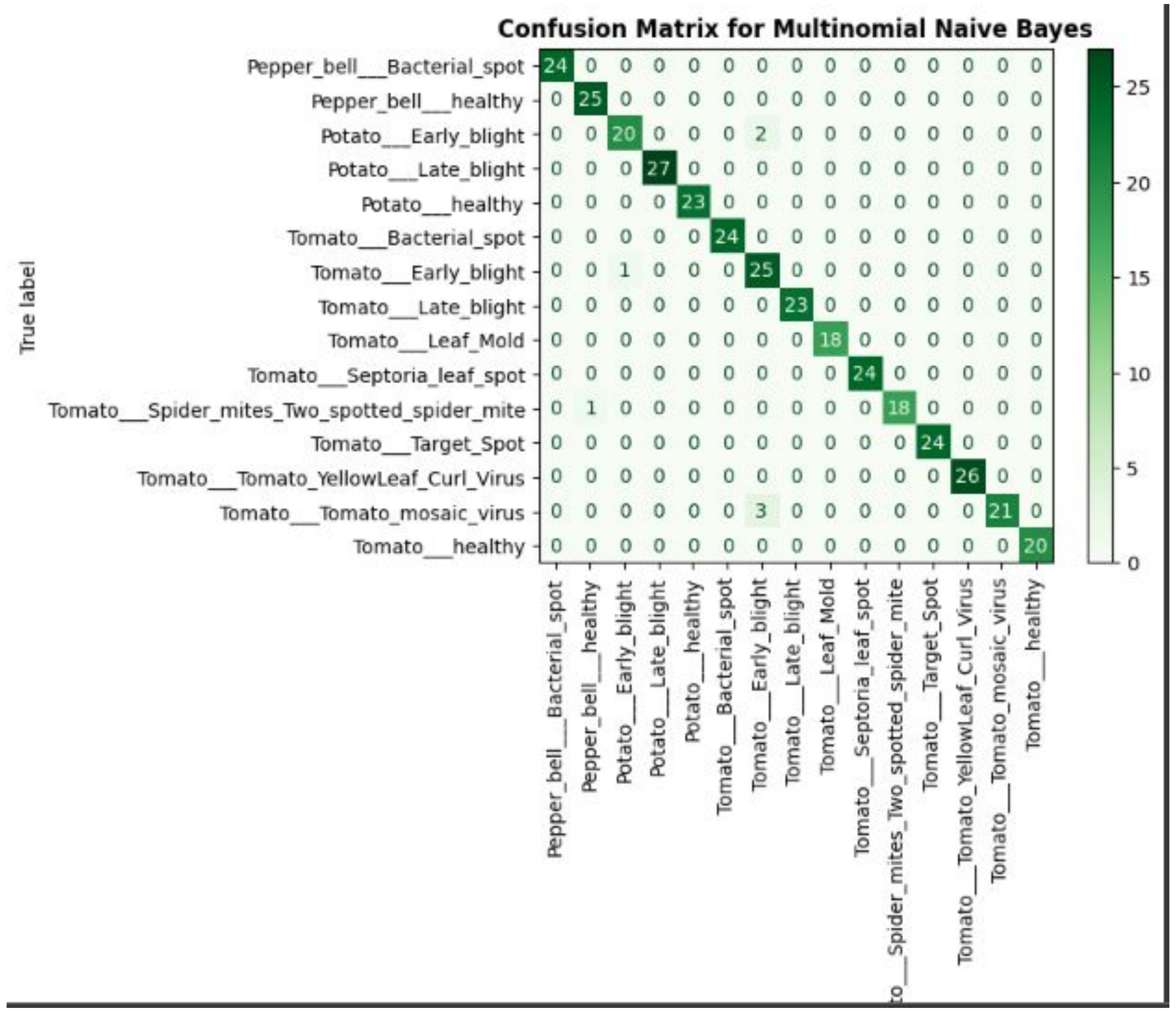
# NLP Performance

## ROC Curve




**Accuracy:** 0.98
**Recall:** 0.98
**Precision:** 0.98
**F1-Score:** 0.98

## Confusion matrix



# Deployment Approach

- Save trained models:
  - CNN: .keras or .h5
  - NLP pipeline: .pkl (e.g. with joblib)
- Possible deployment:
  - Web app (Flask, FastAPI, Streamlit)
  - Mobile app
- User uploads:
  - Leaf photo → CNN model
  - Text description → NLP mode

# Challenges & Future Work

## Challenges:

- ❖ Similar symptoms between diseases
- ❖ Variability in farmer descriptions

## Future work:

- ❖ Expand dataset
- ❖ Use transfer learning (e.g. EfficientNet)
- ❖ Multilingual NLP models
- ❖ Integration with real-time advisory systems

# Thank You

GROUP SEVEN