Advancing Agricultural Health

7th August 2024

W281 Computer Vision (Summer 2024) Final Project

Nikolay, Closson; Maxence, Frenette; Brian, Lin; Agnese, Minazzo; Mana, Vahid

Berkeley
UNIVERSITY OF CALIFORNIA

Objective

Develop image classification system using PlantVillage dataset to accurately identify and classify various crop diseases.

Preprocessing data

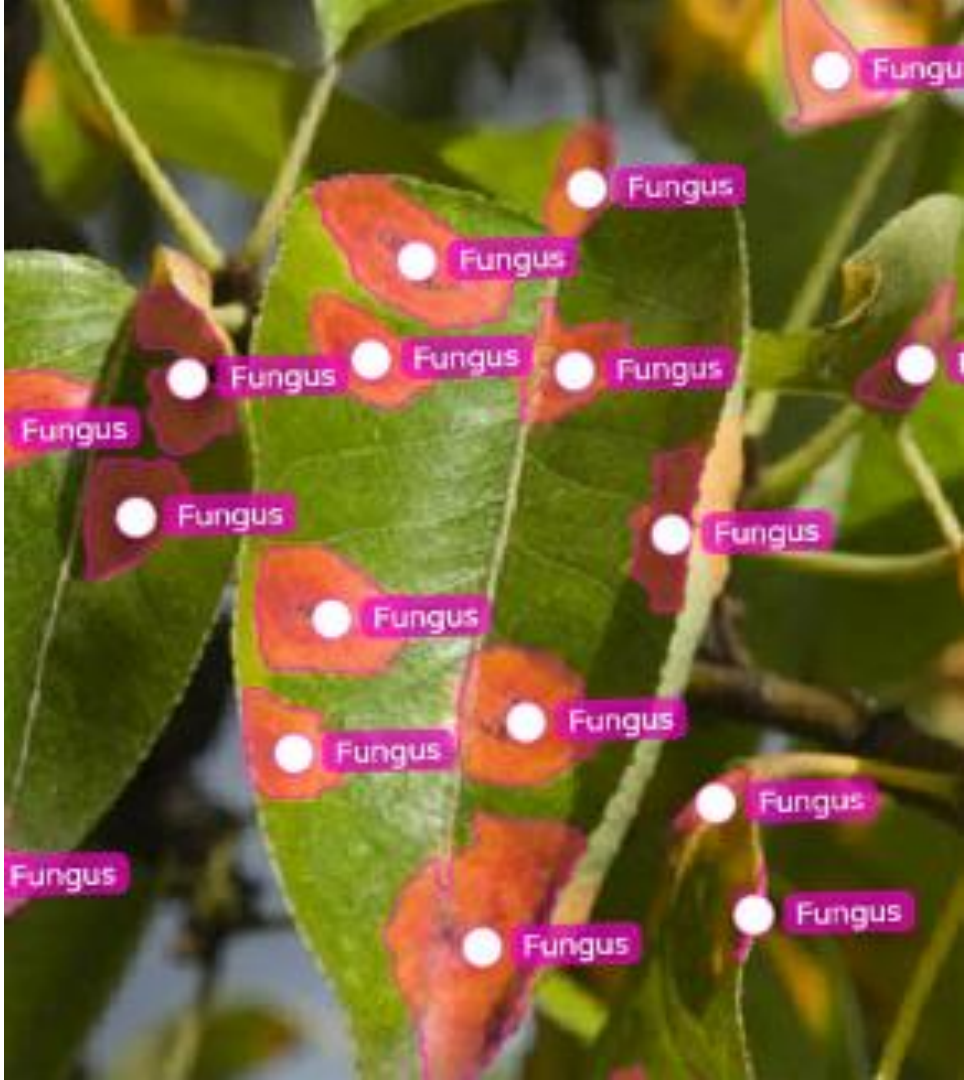
- Right size
- Normalization
- Background

Implement and Evaluate Feature Extraction Methods

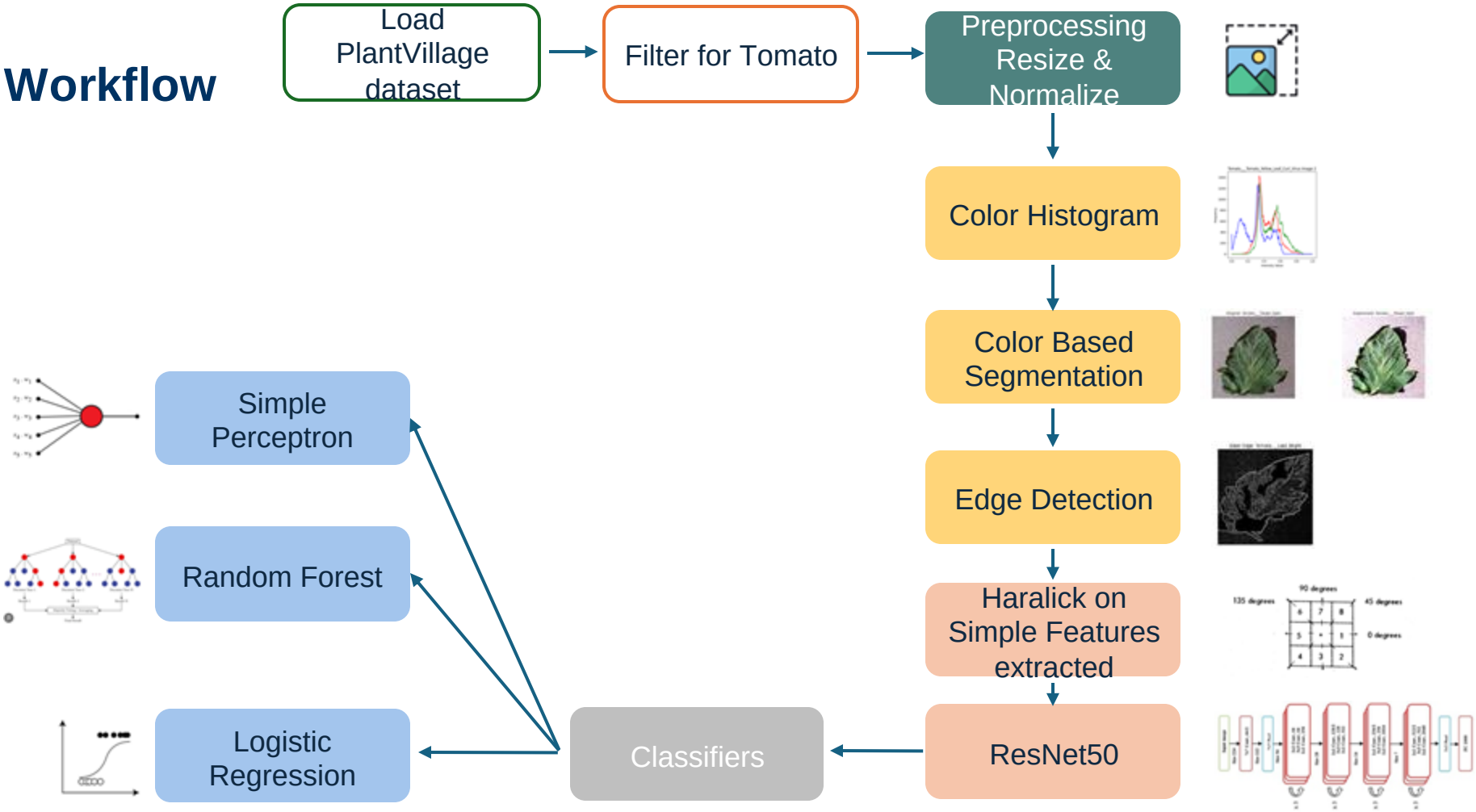
- Simple Features
- Complex Features

Classifiers Training

- Simple Perceptron
- Random Forest
- Logistic Regression

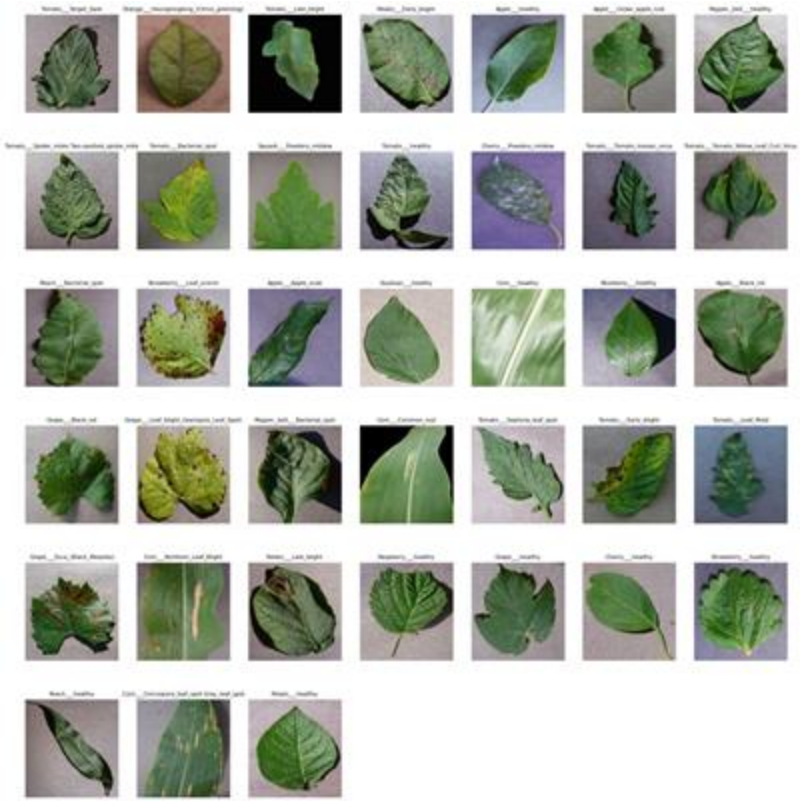


Workflow


Load PlantVillage dataset
Filter for Tomato
Preprocessing Resize & Normalize
Color Histogram
Color Based Segmentation
Edge Detection
Haralick on Simple Features extracted
ResNet50
Classifiers
Simple Perceptron
Random Forest
Logistic Regression

Data

- 54,303 images of both healthy and unhealthy leaves of various crops
- 14 types of crops
- 38 distinct classes
- No null values
- No duplicates

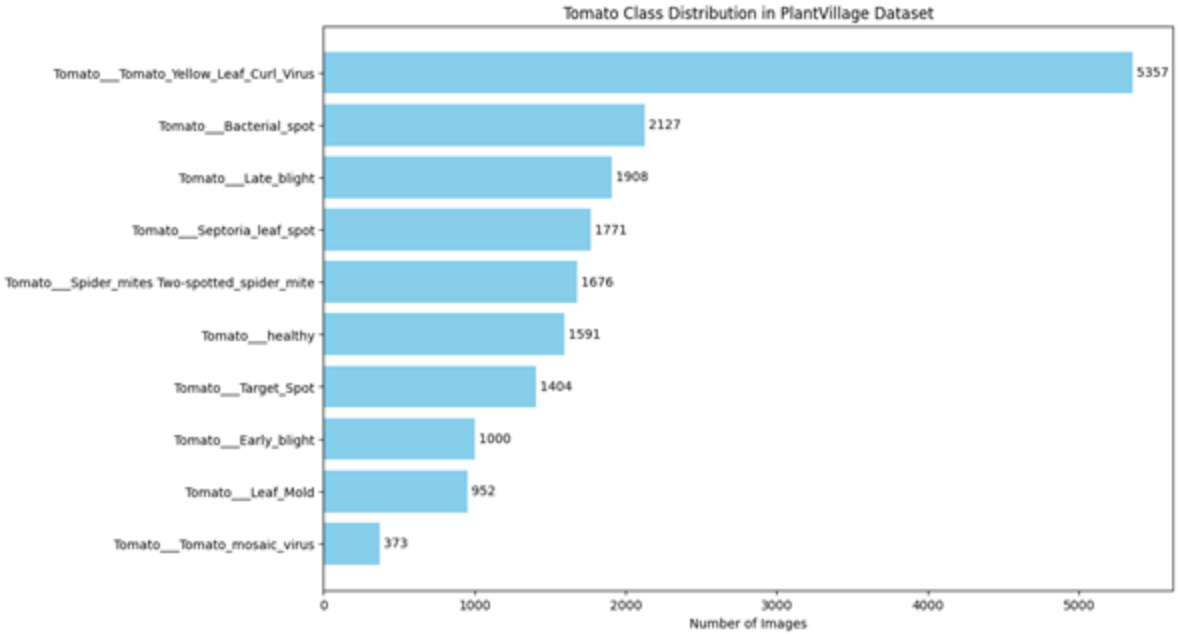


Crop Name	Healthy	Unhealthy	Total
Tomato	1591	16568	18159
Orange	0	5507	5507
Potato	152	2000	2152
Apple	1645	1526	3171
Pepper, Belt	1477	997	2474
Squash	0	1835	1835
Cherry	854	1052	1906
Peach	360	2297	2657
Strawberry	456	1109	1565
Soybean	5090	0	5090
Corn	1162	2690	3852
Blueberry	1502	0	1502
Grape	423	3639	4062
Raspberry	371	0	371

Data

Focus on Tomato

- 18,159 images
- 10 classes
  - unbalanced
- Variation in image size:
  - Min size: 5.07 KB
  - Max size: 54.05 KB



Processing

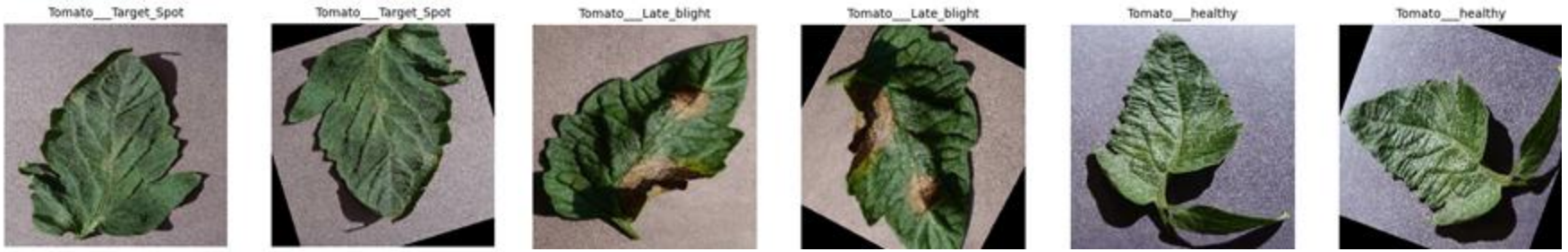
Resize and normalize:
Resized to standard standard dimension of 256x256 pixels and normalizing pixel values to the [0, 1].

Grayscale Conversion:
Using grayscale does not seem a good approach and not effective as it obscures specific color variations crucial for identifying health conditions in plants.

Gaussian filter:
Gaussian filtering reduced noise but smoothed out disease-related features, impacting the visibility of leaves' edges.

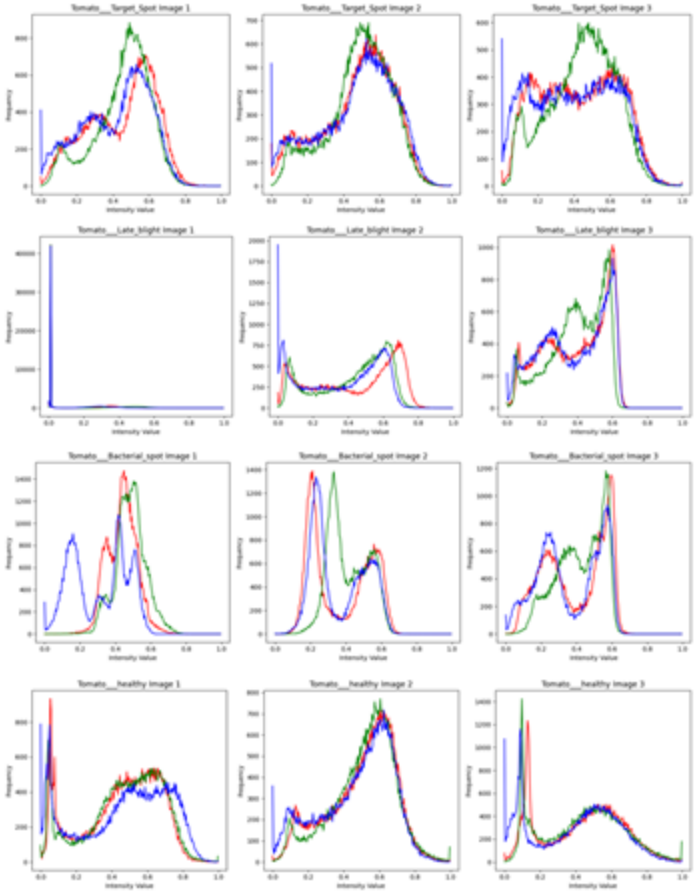
Preprocessing - Rotating images

- Rotation was considered to simulate different leaf orientations, but it introduced black areas in the background, complicating consistency and feature extraction.
- Due to the large dataset and potential computational challenges, the approach was not pursued.



Simple Features Extraction

- **Color histograms** revealed RGB patterns and intensity distributions for various tomato leaf diseases, aiding in their visual identification.
- Different diseases, such as Target Spot, Late Blight, and Bacterial Spot, show characteristic peaks in specific intensity ranges, reflecting unique color variations and symptoms.
- Healthy leaves display smooth color distributions, while diseased leaves show more defined peaks, indicating alterations in natural color due to symptoms like spots, lesions, or discoloration.

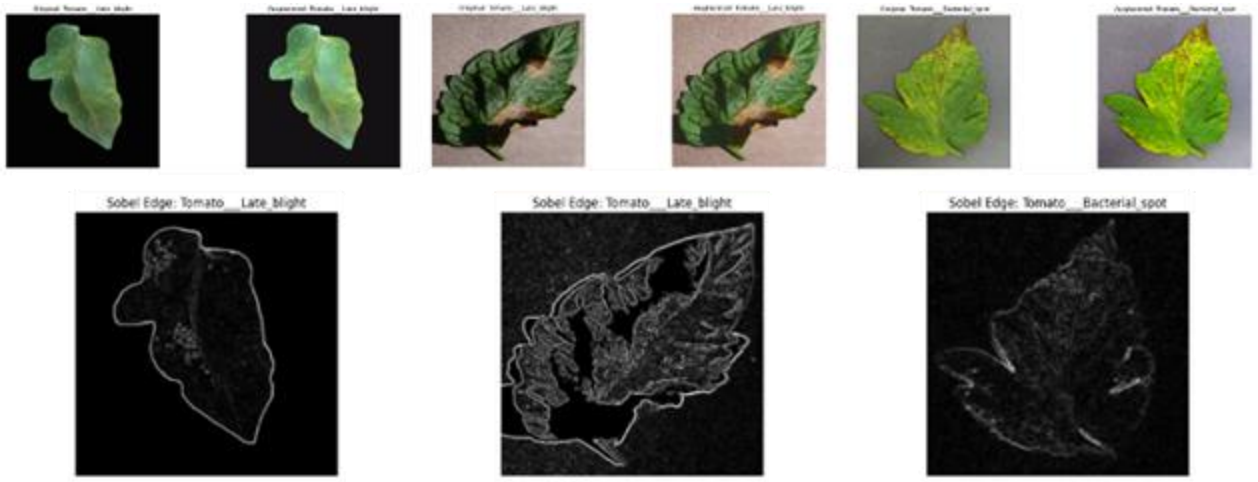


Simple Features Extraction

- **Color based segmentation** highlighted regions of interest by focusing on color ranges associated with disease symptoms, such as red, brown, and yellow areas, while excluding cooler tones, such as blue.
- Lower bound: [0.5, 0.2, 0.0]
  - Red: minimum intensity of 0.5 (ensure some red is present)
  - Green: minimum intensity of 0.2 (to allow yellow and brown tomes)
  - Blue: minimum intensity of 0.0 (allowing for no blue, necessary for both red and yellow)
- Upper bound: [1.0, 1.0, 0.5]
  - Red: maximum intensity of 1.0 (full red intensity).
  - Green: maximum intensity of 1.0 (full green intensity, important for yellow)
  - Blue: maximum intensity of 0.5 (restricting blue to prevent it from overpowering the other colors).

Simple Features Extraction

- **Edge detection** emphasized boundaries of disease.
- **Background** removal effectively masks out the background by retaining only the pixels within the defined color bounds, isolating potential disease-affected areas.



Complex Features Extraction

Haralick features are statistical measures of texture that describe the arrangement and properties of pixels in an image, useful for capturing patterns that might be indicative of disease.

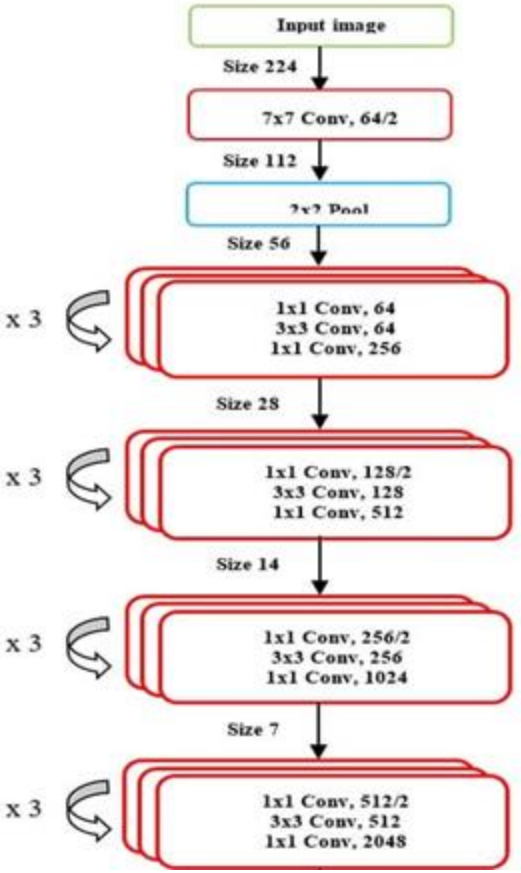
We computed such measures separately for each color channel (although are normally computed in grayscale – color is important for leaves disease) and combined to retain color information, capturing patterns indicative of disease.

Breakdown:

- **Image Conversion to Features**: Each RGB channel of an image is separately processed to generate Haralick features, capturing texture information useful for tasks like identifying plant diseases.
- **Combining Features:** Features from each channel are concatenated into a single 15-element vector, summarizing contrast, dissimilarity, homogeneity, energy, and correlation across RGB channels.
- **Feature Vector:** The 15-element vector provides a summary of the textural characteristics of the image.

Complex Features Extraction

- **ResNet50** -> Pretrained CNN for image classification
- Trained on ~14.2 million images, across over 21,000 class
- Extracted the final 2048 length embedding vector to use for our classifiers



Model Preprocessing

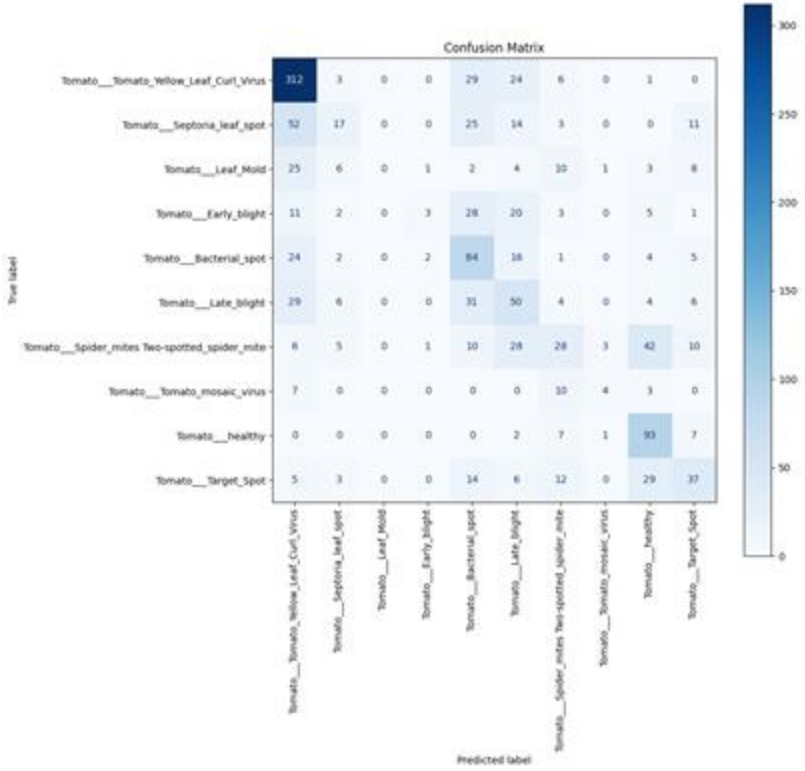
1. Combined the features set
   - Haralick applied to simple features extraction
2. Normalized feature values
3. Encoded the output label
4. Split dataset into train/validation/test (70/15/15)

```
Train set size: 12711
Validation set size: 2723
Test set size: 2725
```

```
Features shape: (2063,)
Features dtype: <dtype: 'float32'>
Label: 6
Label dtype: <dtype: 'int64'>
```

```
Mapping:
0: [UNK]
1: Tomato___Tomato_Yellow_Leaf_Curl_Virus
2: Tomato___Septoria_leaf_spot
3: Tomato___Leaf_Mold
4: Tomato___Early_blight
5: Tomato___Bacterial_spot
6: Tomato___Late_blight
7: Tomato___Spider_mites Two-spotted_spider_mite
8: Tomato___Tomato_mosaic_virus
9: Tomato___healthy
10: Tomato___Target_Spot
```

1. Logistic Regression - Confusion Matrix



Logistic Regression Classifier manage unbalanced data not well.

Data preparation:
Run on a subset of samples.Converted labels to one-hot encoding for multi-class classification.

Model Architecture:
Input Layer: Dense layer with 128 neurons and ReLU activation.
Hidden Layer: Dense layer with 64 neurons and ReLU activation.
Output Layer: Dense layer with softmax activation for multi-class classification.

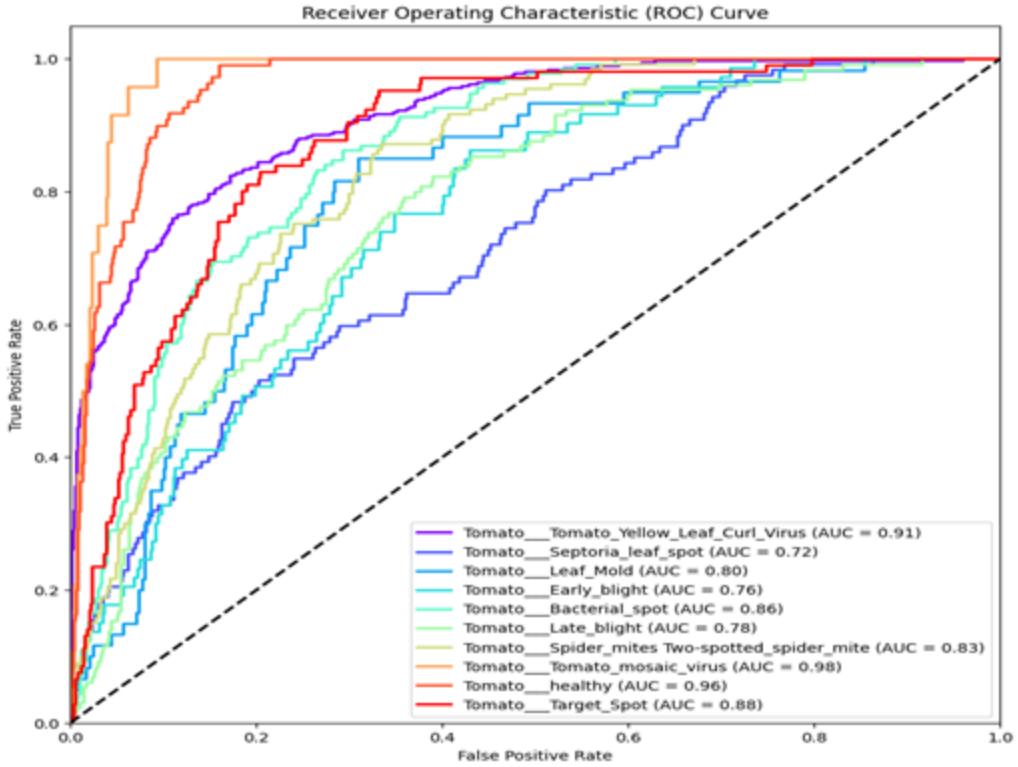
Model Compilation:
Optimizer: Adam.
Loss Function: Categorical Cross Entropy.
Metrics: Accuracy.

Model Training:
Trained the model using the training dataset with validation on the validation dataset.
Used 10 epochs and a batch size of 32.

Hyperparameter Tuning:
Utilized Keras Tuner's RandomSearch to find the best hyperparameters.
Tuning included varying units (32-512), dropout rates (0.0-0.5), and learning rates (1e-4 to 1e-2).
Conducted 20 trials.

The overall accuracy of 49% with weighted average for precision (0.48), recall (0.49) and F1-score (0.44)

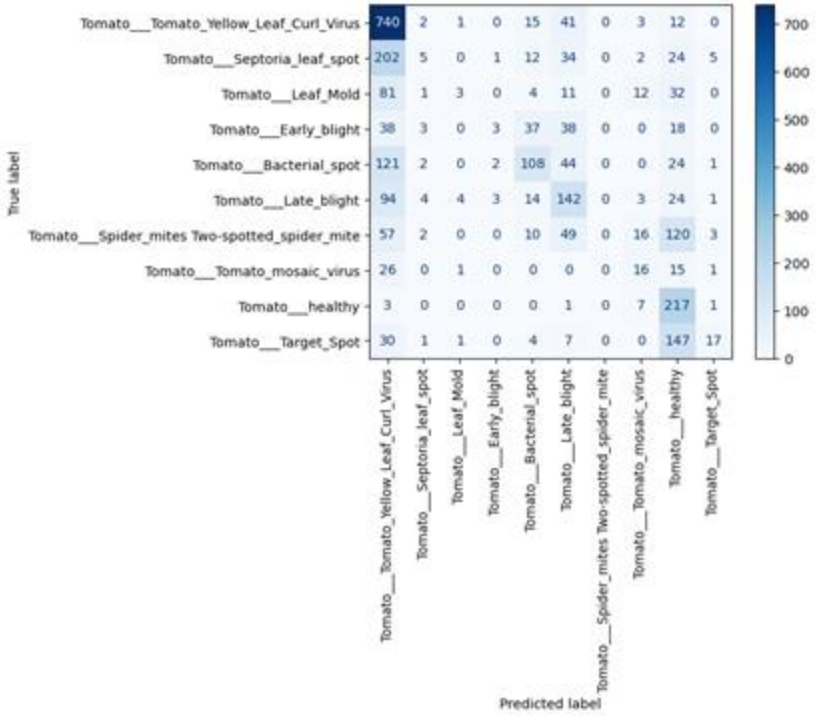
1. Logistic Regression - ROC model



2. Simple Perceptron

- The classifier is a simple perceptron model, which is a type of feedforward neural network.
- Input Layer: 2063 features
- Output layer: 10 neurons
- It is compiled with categorical cross-entropy loss, the Adam optimizer with a learning rate of 0.025, and accuracy as the metric.
- The model is trained for 20 epochs with a batch size of 32, using one-hot. encoded target variables.

2. Simple Perceptron - Confusion Matrix



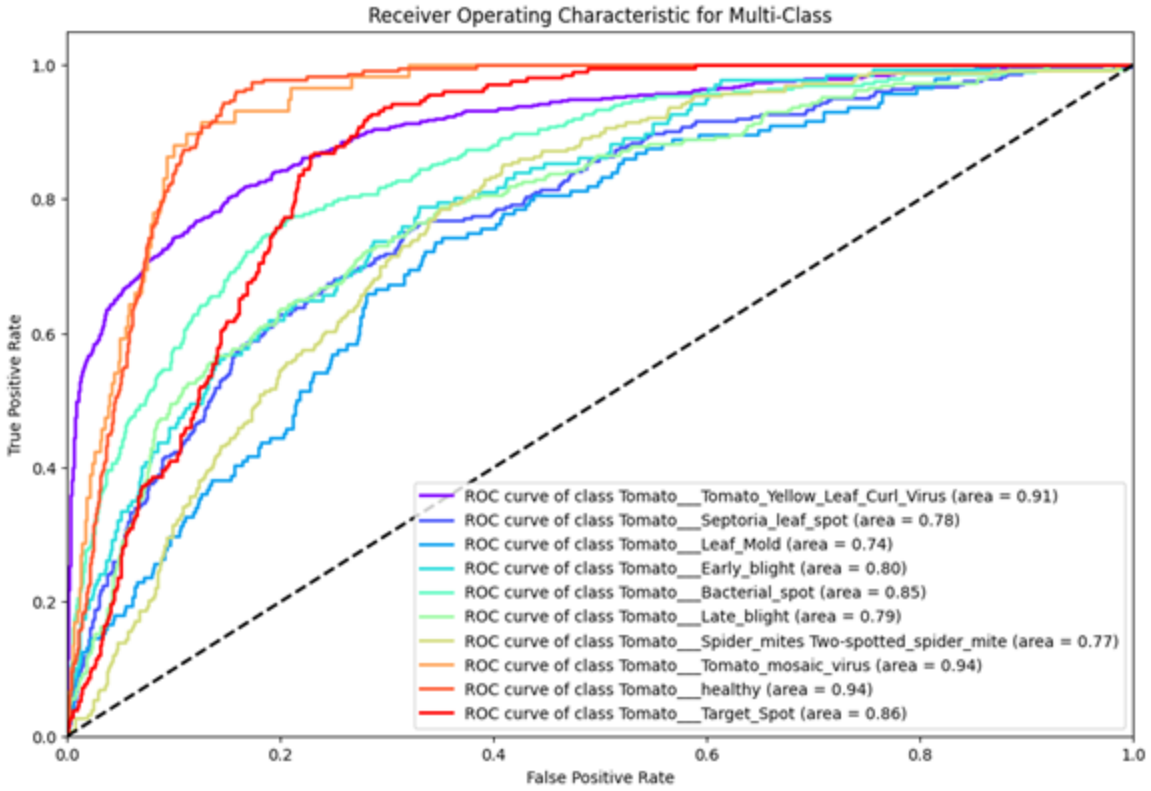
Simple Perceptron - Validation Classification Report

	precision	recall	f1-score
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.57	0.89	0.70
Tomato___Septoria_leaf_spot	0.00	0.00	0.00
Tomato___Leaf_Mold	0.22	0.11	0.14
Tomato___Early_blight	0.32	0.02	0.03
Tomato___Bacterial_spot	0.35	0.66	0.45
Tomato___Late_blight	0.50	0.31	0.38
Tomato___Spider_mites Two-spotted_spider_mite	0.26	0.28	0.27
Tomato___Tomato_mosaic_virus	0.56	0.06	0.11
Tomato___healthy	0.50	0.71	0.58
Tomato___Target_Spot	0.00	0.00	0.00
accuracy			0.47
macro avg	0.33	0.30	0.27
weighted avg	0.37	0.47	0.39

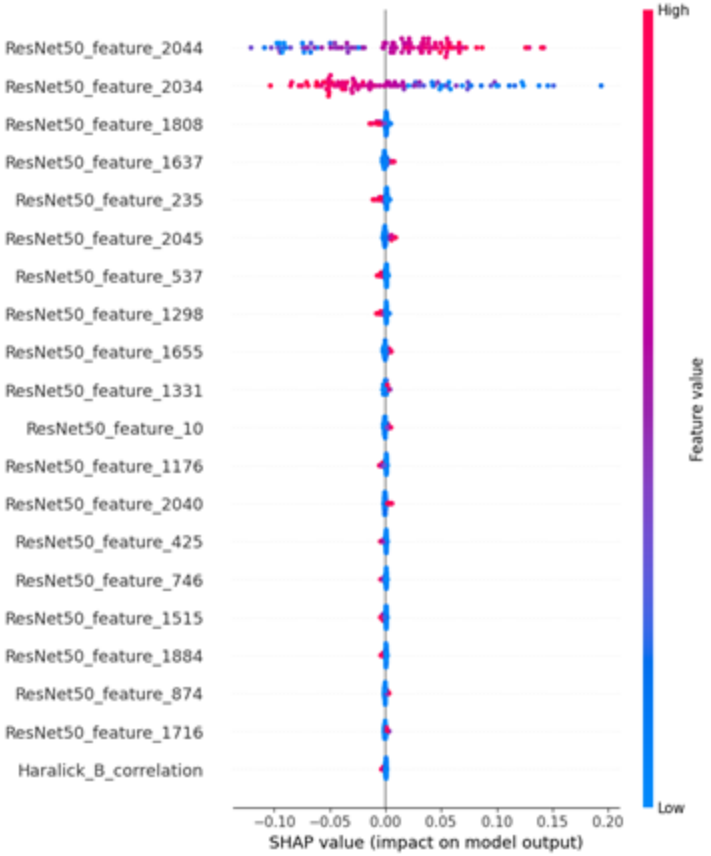
Validation accuracy	0.46
Test Accuracy	0.46

Overall, the model achieves an accuracy of 46%, with weighted average for precision (0.37), recall (0.47) and F1-score (0.39).

2. Simple Perceptron - ROC model



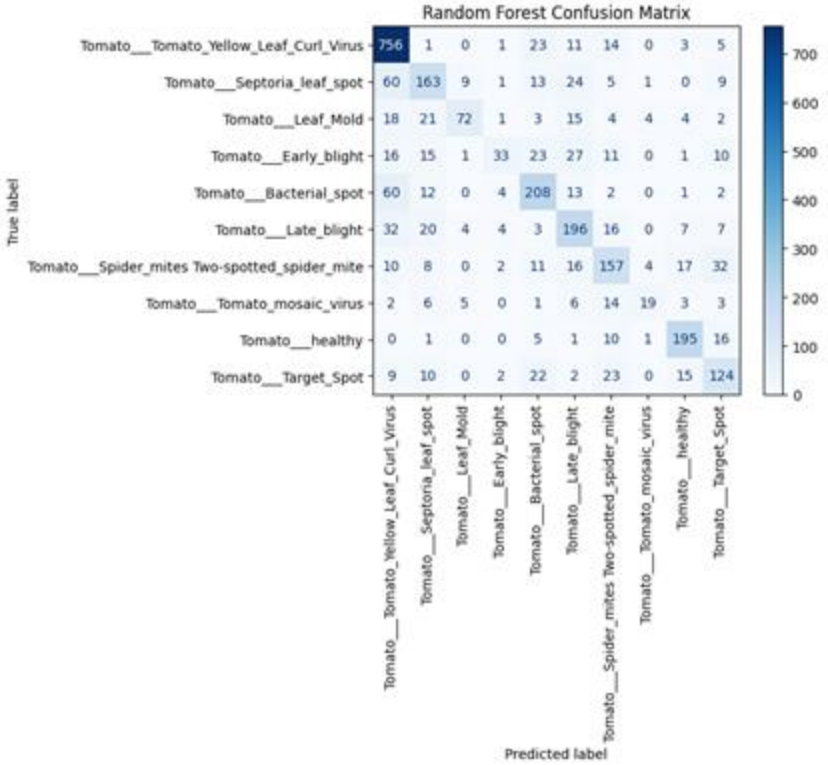
2. Simple Perceptron - ROC best feature



3. Random Forest

- Implemented Random Forest model with 100 decision trees (estimators) to improve robustness.
- Balanced class weights used to manage unbalanced data effectively.
- Trained on the training set (X_*train*, y_*train*) and evaluated on the validation set (X_*valid*)
- Utilization of *GridSearchCV* for hyperparameter tuning:
  - number of estimators, maximum depth, minimum samples split, and minimum samples leaf.

3. Random Forest - Confusion Matrix

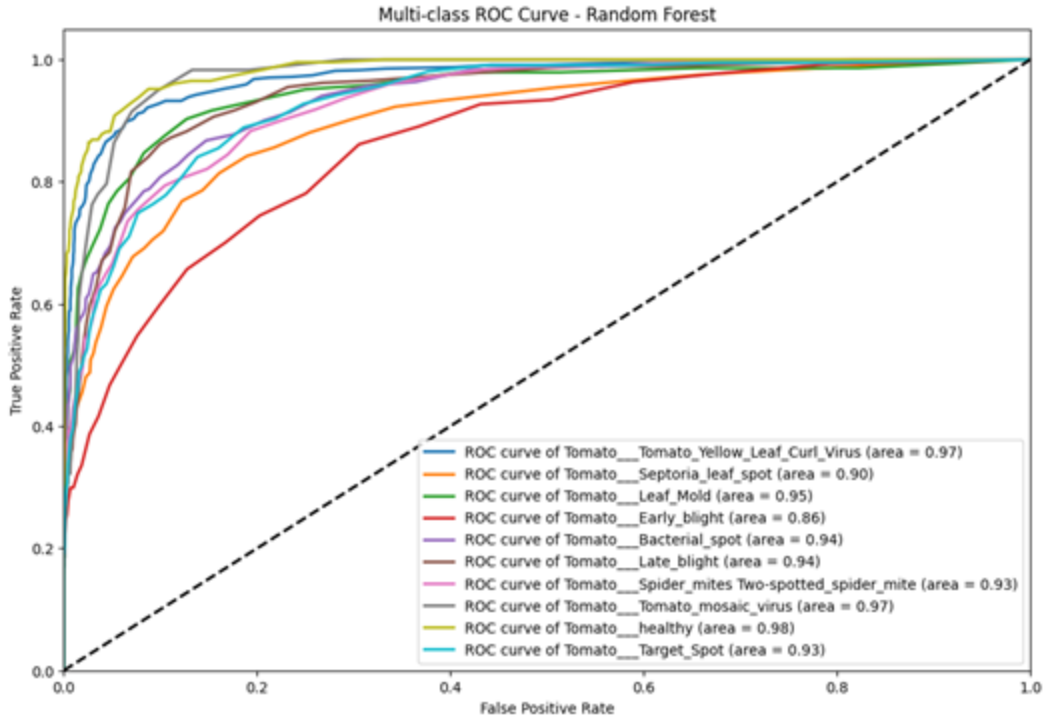


Random Forest - Validation Classification Report

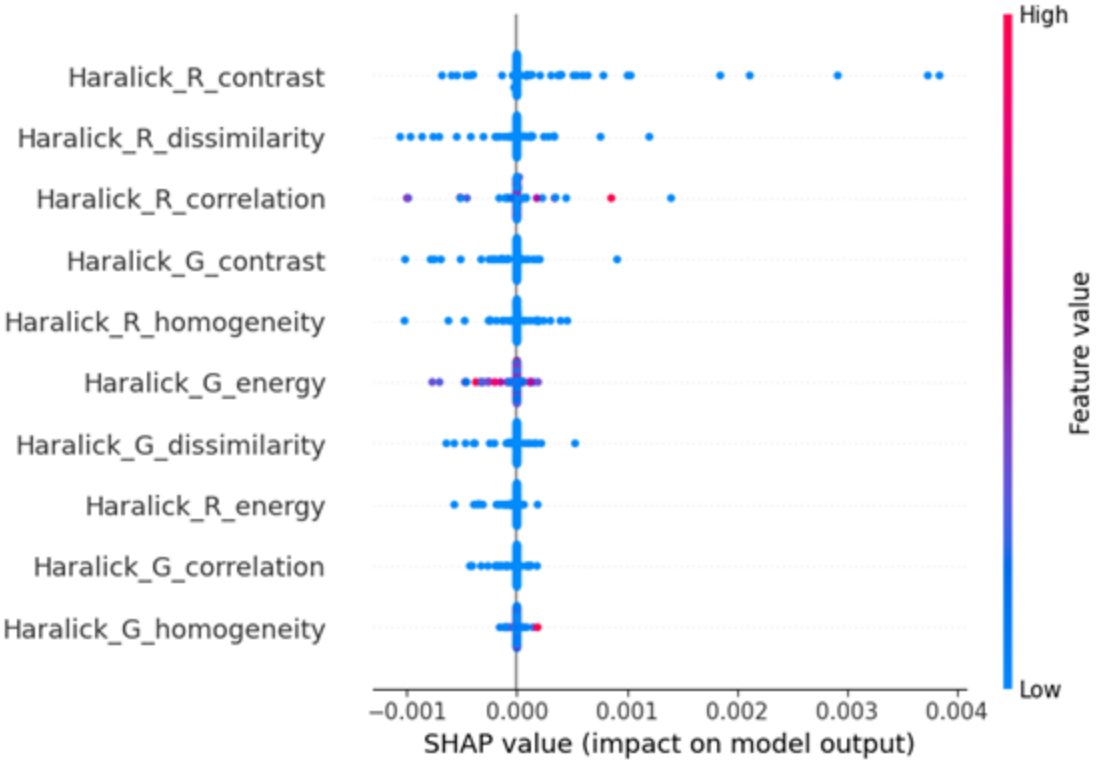
	precision	recall	f1-score
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.79	0.93	0.85
Tomato___Septoria_leaf_spot	0.63	0.57	0.60
Tomato___Leaf_Mold	0.79	0.50	0.61
Tomato___Early_blight	0.69	0.24	0.36
Tomato___Bacterial_spot	0.67	0.69	0.68
Tomato___Late_blight	0.63	0.68	0.65
Tomato___Spider_mites Two-spotted_spider_mite	0.61	0.61	0.61
Tomato___Tomato_mosaic_virus	0.66	0.32	0.43
Tomato___healthy	0.79	0.85	0.82
Tomato___Target_Spot	0.59	0.60	0.59
accuracy			0.71
macro avg	0.68	0.60	0.62
weighted avg	0.79	0.71	0.69

Validation accuracy	0.71
Test Accuracy	0.64

3. Random Forest - ROC model



3. Random Forest - ROC best feature



Result Summary

- Logistic Regression
  - Baseline performance with 49% test accuracy.
  - Struggled with certain classes, e.g., "Tomato_Leaf_Mold" and "Tomato_Early_blight."
  - Overall macro average F1-score of 0.33.
  - ROC analysis showed inconsistent performance across classes (AUC range: 0.72 to 0.96).
- Simple Perceptron Model
  - Validation and test accuracy of 46%.
  - Performed well on "Tomato_Yellow_Leaf_Curl_Virus" but poorly on other classes.
  - Macro average F1-score of 0.27.
  - ROC analysis highlighted varying discriminatory power (AUC range: 0.74 to 0.91).
- Random Forest Classifier
  - Best performance with 71% validation accuracy and 64% test accuracy.
  - High F1-scores across multiple classes, e.g., 0.85 for "Tomato_Yellow_Leaf_Curl_Virus."
  - Macro average F1-score of 0.62.
  - ROC analysis showed high AUC values (0.90 to 0.98) across all classes.
  - Confusion matrix revealed fewer misclassifications compared to other models.

Conclusion

- Combining simple and complex methods enhances accuracy and reliability.
- Provides a scalable solution for automated disease diagnosis.
- Helps mitigate yield losses, improve crop management, and contribute to global food security.

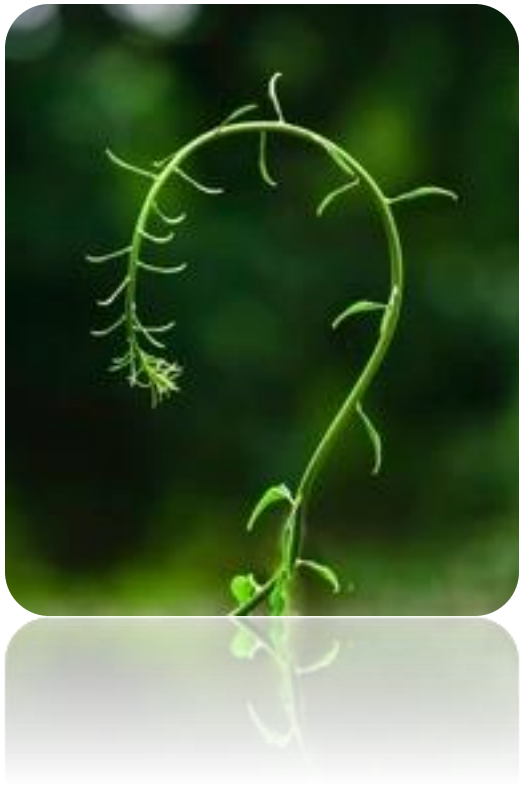
Conclusion

Future Directions

- Dataset Enhancement: Adjust for balanced representation of disease classes; collect images from specific tomato varieties.
- Feature Extraction: Explore adding Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG).
- Advanced Models: Consider leveraging Convolutional Neural Networks (CNNs) or Visual Transformers for improved classification.

Impact

- A scalable solution to mitigate yield losses.
- Enhances crop management and supports global food security.



& A