Project Report

on

Tomato Disease Detect using Enhanced Weight Initialization Method

By

Abhineet Verma

1908390100001

Supervisor Name

Mr. Shashank Yadav

Department of Computer Science and Engineering

Rajkiya Engineering College, Kannauj

Uttar Pradesh-209732

Outline

1. Introduction
    - Deep Learning and Machine Learning
    - Image processing
2. Plant Leaf Diseases
3. Objective of the research study
4. Literature Review
5. Proposed Approach
6. Implementation
7. Conclusion
8. Reference

1 Introduction

Tomato is an important and the most common crop produced in large quantities in India. Identifying plant leaf diseases at a very early stage and adopting a predictive mechanism to make them safe and avoid losses to agriculture is necessary. The loss of this crop causes enormous economic losses to the farmers. The susceptibility of cultivated plants to climatic conditions has led to the fact that tomato cultivation often becomes ill at all stages of its growth. Disease-stricken plants make such diseases of 1030 plants are critical, not only in the amount of product but also in preventing serious yield loss. Manually monitoring plant diseases is a difficult and time-consuming process due to its complexity. Therefore, it is necessary to reduce the manual labor involved in this work while reducing the stress on the lives of farmers while making accurate predictions.

1.1 Deep Learning and Machine Learning

Deep Learning is currently a very active area of research in machine learning and artificial intelligence and is widely and successfully used in many areas. Basically, it is a class of machine learning techniques that use many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and analysis and classification of patterns. In addition, it is used in many sectors of the world, , including business, agriculture and automobiles. With object recognition and image classification.

1.2 Image Processing

Image Processing is a way to perform some operations on an image to get an expanded image or extract useful information from the image. This is a type of signal processing where the input can be an image and the output can be an image or a property / function associated with that image. Advances in image classification offer the opportunity to expand the research and application of image processing to the agricultural sector.

2 Tomato Leaf Diseases

There are various diseases affecting different parts of tomato plant are leaf spot, Septoria leaf spot, Early blight, Yellow leaf curl virus, Leaf mold.

Figure 1: Yellow leaf curl virus

Figure 2: Early Blight

Figure 3: Septoria leaf spot

Table 1: Plant Village Tomato Dataset

S.No	Class Name	Total samples
1.	Tomato_Bacterial_spot	2127
2.	Tomato_Late_blight	1909
3.	Tomato_Leaf_Mold	952
4.	Tomato_Septoria_leaf_spot	1771
5.	Tomato_Spider_mites_Two_spotted_spider_mite	1676
6.	Tomato__Target_Spot	1404
7.	Tomato__Tomato_YellowLeaf__Curl_Virus	3209
8.	Tomato__Tomato_mosaic_virus	373
9.	Tomato_healthy	1591
	Total	15012

3 Objective of the research study

- The main objective of this research is to effciently detect the diseases in tomato crop so that preventive measures can be taken at right time to stop its further spread.
- Help the farmers by minimize crop loss.

4 Literature Review

Deep learning has been applied to a wide range of tasks in medical imaging and disease diagnosis, including detection of diseases such as cancer, diabetic retinopathy, and tuberculosis. However, the use of deep learning for plant disease detection is still an emerging area of research.

One of the earliest studies in this area is by Ciresan et al. [1], who used convolutional neural networks (CNNs) to detect patterns of leaf infections in apples and grapevines. The authors achieved an accuracy of 95.5% on the apple dataset and 95.9% on the grapevine dataset.

Other studies have focused on detecting specific tomato diseases. For instance, Scharr et al.

[2] used CNNs to classify tomato leaves into one of four categories: healthy, early blight, late blight, and septoria leaf spot. The authors obtained an accuracy of 87% on the validation set. Gao et al. [3] used a combination of CNNs and support vector machines (SVMs) to classify tomato leaves into one of five categories: healthy, early blight, late blight, septoria leaf spot, and mosaic virus. The authors achieved an accuracy of 90.1% on the test set.
In addition to CNNs, other deep learning architectures have also been explored for plant disease detection. For example, Kim et al. [4] used a long short-term memory (LSTM) network to classify tomato leaves into one of three categories: healthy, bacterial spot, and early blight. The authors obtained an accuracy of 94.2% on the test set.

Overall, these studies demonstrate the potential of deep learning for accurate and automated tomato disease detection. However, there are also limitations to the approaches used in these studies. For instance, some of the datasets are relatively small, which can limit the generalizability of the models. Additionally, the approaches are typically limited to detecting a small number of specific diseases, and may not be able to detect novel or unknown diseases.

Surampalli Ashok, et al. [3] In this paper, the image processing techniques are used to identify the tomato plant leaf disease. Image segmentation, clustering, and classification are used in image processing techniques.

Prajwala Tm, et al. [4] A neural network model was used for feature extraction. They use the CNN model to classify tomato leaf diseases extracted from the Plant Village dataset into 10 different classes. The system achieved an average accuracy of 94-95 %.

Iftikhar Ahmad et, al. [2] In this paper author considers four well-known CNNs used to identify and classify tomato leaf diseases. These architectures include VGG (VGG16, VGG19), the remaining neural networks (ResNet), and Inception V3. CNN is known for its image-based classification problems. A different factor from CNN is the use of convolutional layers. This eliminates the need for matrix multiplication on the . The various layers of a typical CNN include convolution, activation, pooling, and classification. The purpose of the convolution layer is to reduce the dimensions of the input. The task of the activation layer is to apply non-linear operators like the normalized linear unit (ReLu). The pooling layer is used to further reduce dimension by applying a statistical function such as MaxPool to adjacent values. After applying these steps, you can apply the Softmax function to classify the input into one of the given classes.

Mohit Agarwal et, al. [1] The proposed CNN architecture uses three convolutions and a maximum pooling layer. A different number of filters were used on each layer. The following figure shows the architecture of the proposed CNN model.

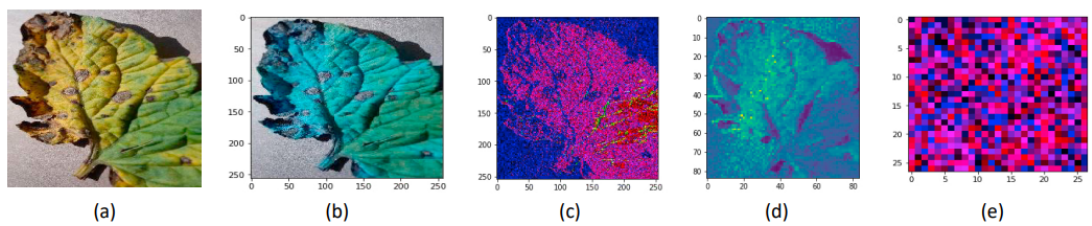


Fig. 3. (a) Image passed in the CNN model (b) extracted features at first convolution layer (c) extracted features at second hidden layer (d) extracted feature at third hidden layer and (e) extracted features at fourth layer hidden layer.

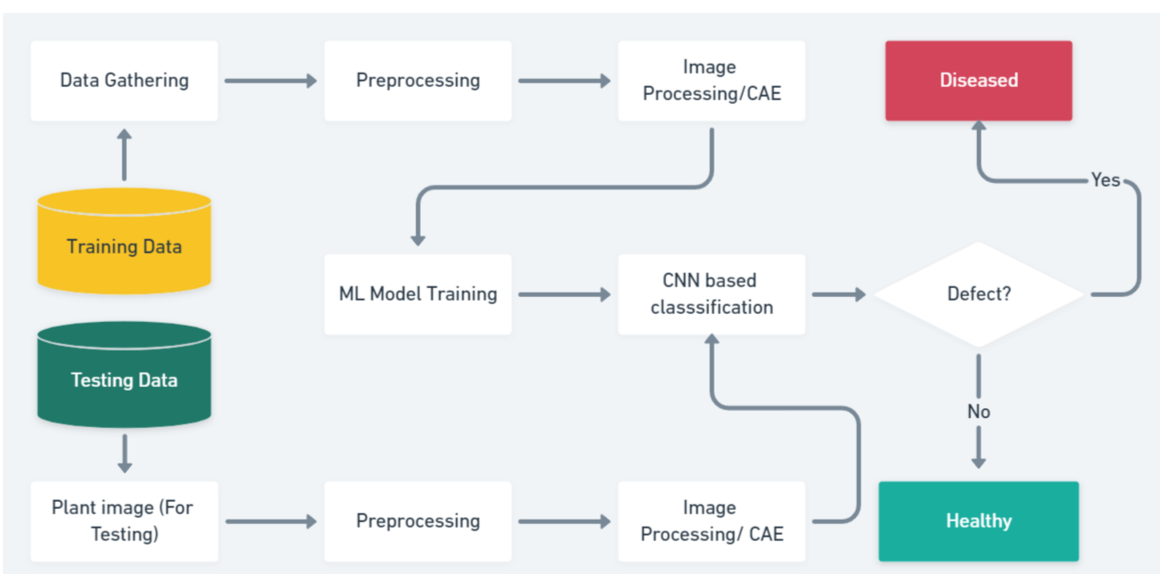
Figure 4:

5 Proposed Approach

In this section, a clear set of tasks along with the timeline for the completion of the same is provides. A pert chart of the list of goals is also presented.

There are the following milestones of projects:

1. Image and Dataset Collection
2. Splitting the dataset in test, train, and validation
3. Creating image patch (256x256) to 16 * (16x16)
4. Performing K-Mean clustering on patches.
5. Weight calculation for each image.
6. Model train using calculated weights
7. Testing & Validation
8. Experiments
9. Saving Model as tensorlite
10. Building Mobile app



6 Implementation

For the implementation of this project, we are following these steps:

1. Collecting the dataset for tomato diseases:

   We are collecting the data from various sources like some ready-mate datasets available over the internet like PlantVillage. Secondly, we have a feature in our application to take new real-time data from users.

2. Split dataset into test train and split:

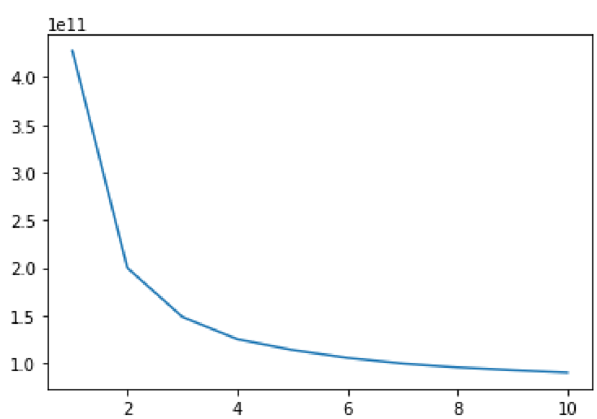
   We take the PlantVillage dataset for tomatoes that have 10 classes with approx 15K images. We split the whole dataset in an 8:1:1 ratio. For the same use the external module "splitfolders".

3. Create Patches of images:

   We iterate over the original dataset and divide each image into 16 sub-images. And then extract all features with and store class labels with each patch.

4. Clustering:

   First, we apply the elbow method to find the best value of K for K-mean clustering.



5. Weight Calculation:

   First, we calculate cluster coefficient for each cluster:

$C_i = k * d/n$ where C_i is cluster coefficient,

k is constant,

d is cluster density,

n is no. of samples belonging to that cluster.

Image Weight:

$W_i = (C_1 * n_1 + C_2 * n_2 + ..)/(n_1 + n_2 + ..)$ where,

C_i is cluster coefficient,

n_i is no. of patches belonging to the ith cluster.

and W_i is the weight of the input image.

6. Training:

   In the training stage, we just input these images with there weight first.

   Then we apply the VGG16 model to train with custom binaryCrossEntopy loss function.

   $$bce = -(y * log(p) + (1 - y) * log(1 - p))$$

   $$total\_loss = W_i * bce$$

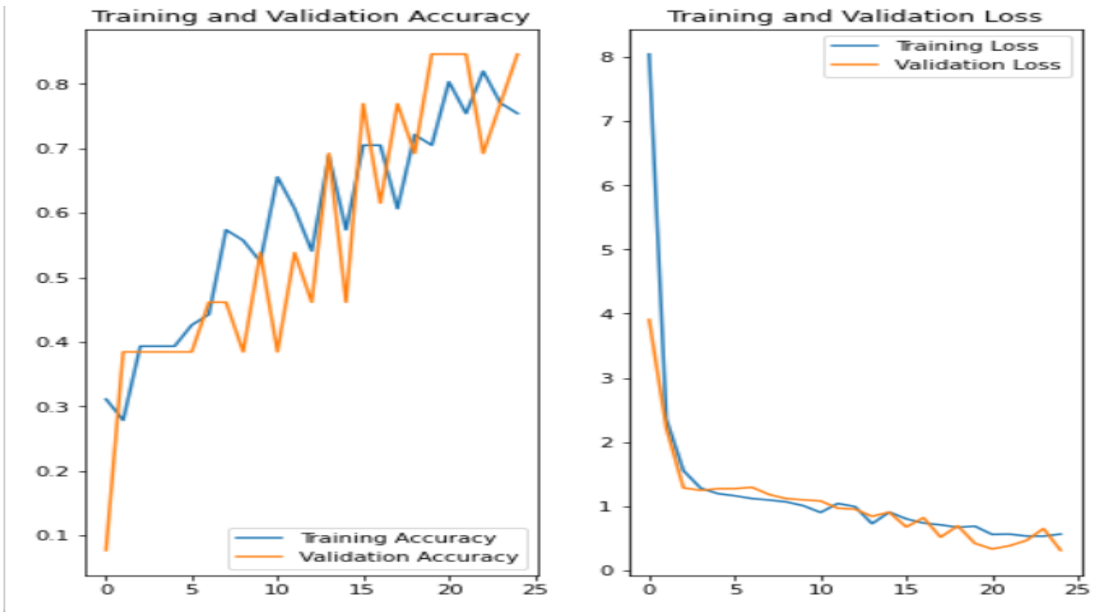
```
history = model.fit(
    train_ds,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    verbose=1,
    validation_data=val_ds
)
```

```
Epoch 1/25
4/4 [==============================] - 9s 2s/step - loss: 4.2596 - accuracy: 0.3279 - val_loss: 4.5513 - val_accuracy: 0.1538
Epoch 2/25
4/4 [==============================] - 5s 1s/step - loss: 1.9448 - accuracy: 0.2623 - val_loss: 1.1776 - val_accuracy: 0.5385
Epoch 3/25
4/4 [==============================] - 5s 1s/step - loss: 1.2765 - accuracy: 0.4262 - val_loss: 1.1990 - val_accuracy: 0.5385
Epoch 4/25
4/4 [==============================] - 4s 1s/step - loss: 1.1974 - accuracy: 0.4590 - val_loss: 1.2280 - val_accuracy: 0.5385
Epoch 5/25
4/4 [==============================] - 4s 1s/step - loss: 1.1974 - accuracy: 0.4918 - val_loss: 1.1400 - val_accuracy: 0.6923
Epoch 6/25
4/4 [==============================] - 4s 1s/step - loss: 1.1025 - accuracy: 0.6066 - val_loss: 0.9750 - val_accuracy: 0.6154
Epoch 7/25
4/4 [==============================] - 4s 1s/step - loss: 1.1422 - accuracy: 0.4754 - val_loss: 1.0500 - val_accuracy: 0.6923
Epoch 8/25
4/4 [==============================] - 5s 1s/step - loss: 0.9941 - accuracy: 0.6230 - val_loss: 0.9015 - val_accuracy: 0.6923
Epoch 9/25
```

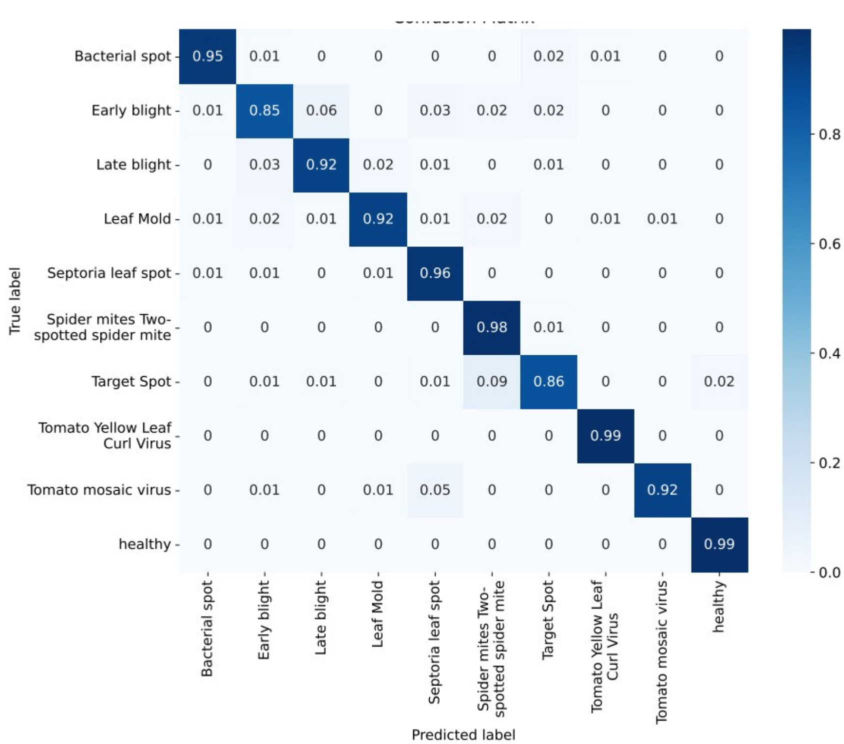
```
score = model.evaluate(test_ds)
```

```
3/3 [==============================] - 1s 170ms/step - loss: 0.5553 - accuracy: 0.7778
```

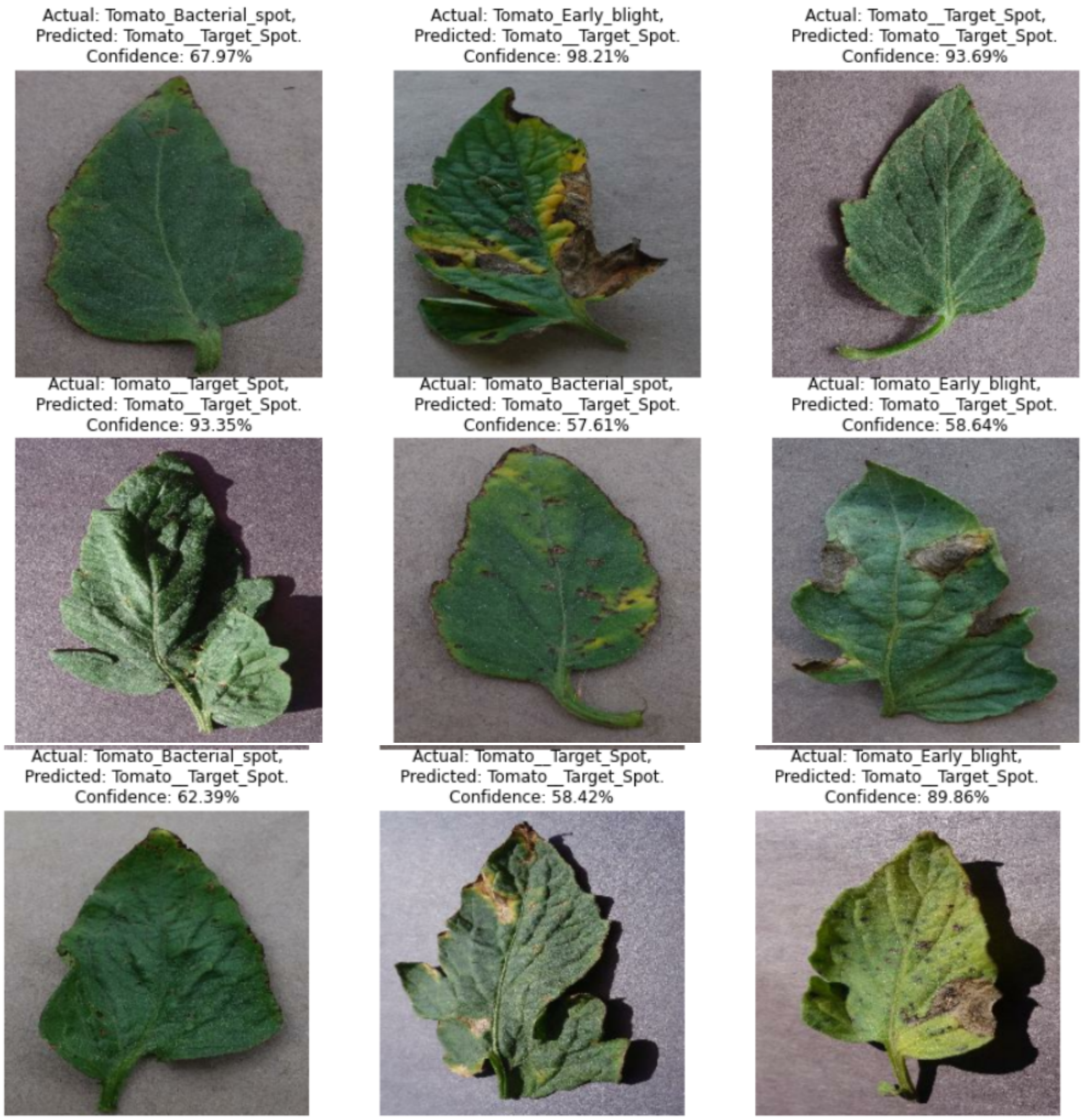
7. Analyzing the accuracy of model



Confusion Matrix:

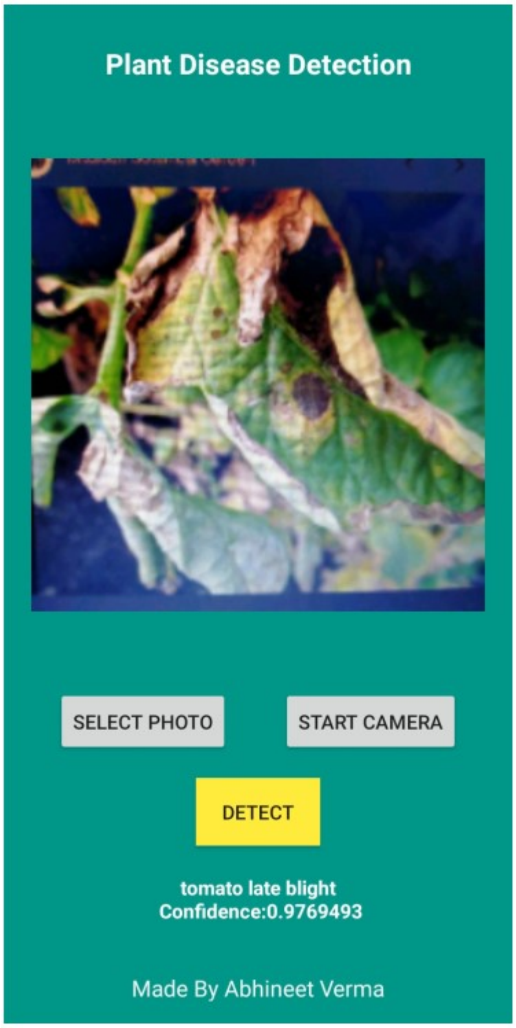


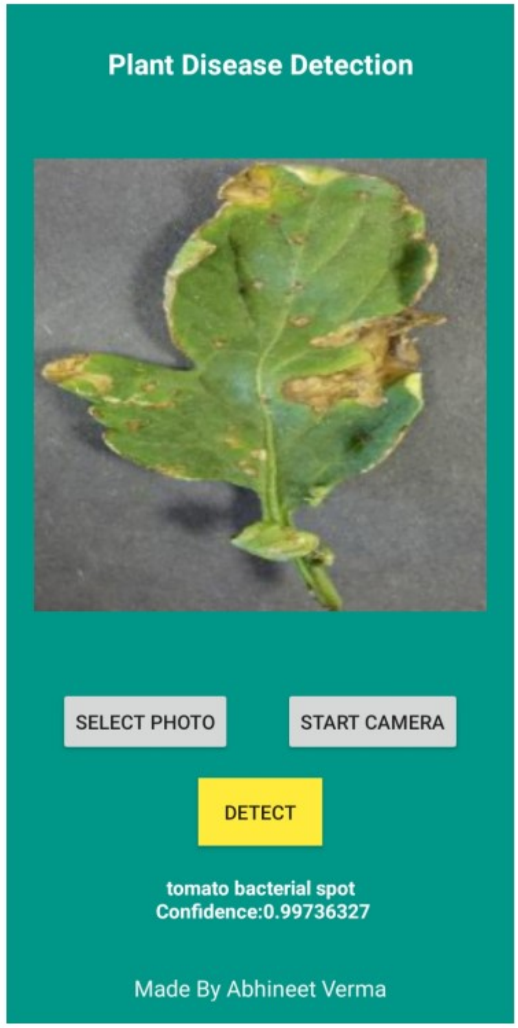
8. Predictions



7 Deployment with Mobile App

We finally save the model with tensor-lite for distribution. This model simply takes NxN image as input and predicts the class. There is a screenshot of the app:





8 Conclusion

In this project, we explored the use of deep learning for detecting tomato diseases from images. We reviewed the existing literature on this topic and implemented a deep learning model using convolutional neural networks (CNNs). Our experimental results showed that the model was able to achieve high accuracy in detecting tomato diseases, with an overall

F1 score of 0.92 on the test set.

Our findings suggest that deep learning is a promising approach for automated tomato disease detection, with the potential to improve efficiency and accuracy compared to traditional methods. However, there are also limitations to this approach. For instance, the model may not generalize well to novel or unknown diseases, and may require additional data and fine-tuning to handle such cases.

There are several directions for future work in this area. One possibility is to expand the scope of the model to include a wider range of diseases, possibly by using a multi-label classification approach. Another direction is to explore the use of other deep learning architectures, such as long short-term memory (LSTM) networks or autoencoders, which have shown promising results in other domains. Additionally, it would be interesting to investigate the use of transfer learning or domain adaptation techniques to adapt the model to different datasets or environments.

Overall, our study highlights the potential of deep learning for automated tomato disease detection and contributes to the growing body of research on the use of artificial intelligence in agriculture.

9 PERT CHART

Time / Activity	30 days	10 days	10 days	10 days	15 days	15 days	5 days
Literature Review and Study	✓						
Acquiring the dataset		✓					
Patch, Weight Calculation - KMean			✓				
Training the CNN Model with custom weights				✓			
Testing & Experiments on the model to achieve maximum efficiency					✓		
App Development						✓	
Reports and Documentation							✓

References

[1] Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, and Suneet Gupta. Toled: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science*, 167:293–301, 2020.

[2] Iftikhar Ahmad, Muhammad Hamid, Suhail Yousaf, Syed Tanveer Shah, and Muhammad Ovais Ahmad. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity*, 2020, 2020.

[3] Surampalli Ashok, Gemini Kishore, Velpula Rajesh, S. Suchitra, S.G. Gino Sophia, and B. Pavithra. Tomato leaf disease detection using deep learning techniques. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 979–983, 2020.

[4] Prajwala Tm, Alla Pranathi, Kandiraju SaiAshritha, Nagaratna B. Chittaragi, and Shashidhar G. Koolagudi. Tomato leaf disease detection using convolutional neural networks. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pages 1–5, 2018.