



# TOMATO LEAF DISEASES CLASSIFICATION SYSTEM USING DEEP GENERATIVE ADVERSARIAL NETWORK

Aliyu O. Zibril<sup>1,2\*</sup>, Ayodeji Samuel Makinde<sup>1</sup>, Omaji Samuel<sup>1</sup>

<sup>1</sup>Department of Computer Science, Edo State University Uzairue, Edo State, Nigeria,

<sup>2</sup>Department of Computer Science, Auchi Polytechnic, Auchi, P.M.B 13, Edo State, Nigeria.

\*Corresponding email: [zibril21aliyu@edouniversity.edu.ng](mailto:zibril21aliyu@edouniversity.edu.ng)

---

## ABSTRACT

Tomatoes are the major source of vegetables for the world population. Tomatoes plant diseases contribute to production loss, which can be tackled with continuous monitoring. Deep Neural Networks (DNNs) have shown remarkable success in various computer vision tasks, including image classification. However, DNNs are vulnerable to adversarial attacks, which can result in misclassification of input images. Therefore, a deep learning-based automatic disease classification system using artificial intelligence (AI) and the creation of artificial images by deep generative adversarial networks (DGANs) on tomato leaves to improve the DNNs for classification was introduced. The main goal of this study is to generate realistic images of tomato leaf diseases thereby enhancing the image quality and using DNN to classify the tomato leaf diseases. This study used a dataset of tomato leaf images with different diseases (Tomato Leaf Spot, Early Blight, Late Blight, Bacterial Spot, and Target Spot) and healthy tomato leaves as the input for the deep generative adversarial network (DGAN). In this work, a cutting-edge classification model was used to classify diseases in tomato leaves. The software components include python programming language, while the hardware component includes a high-processing computer system due to the amount of memory resources required to train the model. In addition, a graphics processing unit (GPU) was used to accelerate the experiment. The results show that the proposed model achieves a high performance in the classification of tomato leaf diseases (Tomato Leaf Spot, Early Blight, Late Blight, Bacterial Spot, and Target Spot), with an accuracy of 97% in the training dataset.

**Keywords:** *Tomato leaf disease, generative adversarial networks, neural network, deep learning, prediction*

---

## 1.0 Introduction

Tomato (*Lycopersicon Esculentum*) grows on almost any well-drained soil. The tomato originated in the Andean region of South America. According to the Food and Agriculture Organization Statistics (FAOSTAT), in the past fifty years, tomatoes have become one of the most important and extensively grown horticultural crops throughout the world (Breitenreiter *et al.*, 2021). Currently, it is the world's second most cultivated vegetable crop after the potato, with approximately 181 million tonnes (Canton, 2021). Tomato is one of the most nutritive crops all over the world, whose cultivation and level of production have a crucial impact on the development of an agricultural economy. Tomato not only contains plenty of nutrition but also has pharmacological effects, which keep people away from diseases such as hypertension, hepatitis, gingival bleeding, and so on (Patel *et al.*, 2020). On account of the wide



use of tomatoes, the demand for tomatoes is also rising. Statistics show that more than 80 percent of tomato production is produced by small farmers (Zemouri *et al.*, 2019).

Tomatoes are highly vulnerable to various diseases that can significantly reduce crop yields. Early detection of these diseases is critical for ensuring global food security. Bacteria, a major plant pathogen, can infiltrate plants through wounds, such as insect bites or pruning, as well as natural openings like stomata. Leaf Mold, particularly Late Blight disease caused by mold, can lead to dark, uneven blemishes on leaves and stems, causing plant weakness and defoliation. Tomato Leaf Spot, a fungal disease characterized by dark circular spots with yellow halos, can weaken plants, hamper photosynthesis, and result in leaf loss. Control methods for these diseases include crop rotation, sanitation, and using resistant cultivars.

Early Blight, caused by *Alternaria Solani*, manifests as brown concentric rings with a target-like pattern on lower tomato leaves, which then enlarge, leading to leaf withering and premature defoliation. Proper cultural practices, fungicides, and adequate plant nutrition are crucial for controlling this fungal disease. Bacterial Spot is a highly destructive disease triggered by *Xanthomonas* spp. Bacteria (Gadade and Kirange, 2022). It leads to the development of dark, raised lesions with a water-soaked appearance on leaves, stems, and fruits. This disease causes significant yield losses, diminishes fruit quality, and even poses a risk of plant death. Effective control measures include using pathogen-free seeds, applying copper-based sprays, and maintaining good sanitation practices (Ontiveros *et al.*, 2022).

Target Spot is a foliar disease caused by the fungus *Corynespora Cassiicola* (Chiu *et al.*, 2021). Initially, it manifests as small, dark spots on leaves, which progressively enlarge and take on a distinctive concentric ring pattern resembling a target (Fazari *et al.*, 2021).

**Scope of the study:** This study focuses on five common classes of unhealthy tomato leaves frequently found in West Africa, namely: Tomato Leaf Spot, Early Blight, Late Blight, Bacterial Spot, and Target Spot.

**Plant Disease Recognition and Classification System:** While small and local farmers rely on traditional manual detection methods to identify crop pests and diseases, these factors remain significant in impacting the growth of tomato crops, emphasizing the need to study their identification (Hammou *et al.*, 2022). With the continuous development of the Internet, image-based disease identification has seen huge adoption in computer vision applications. People use efficient image identification technology to process images, which can improve the efficiency of image recognition, reduce cost, and improve the accuracy of recognition (Dandawate *et al.*, 2022). Artificial intelligence (AI) and machine learning (ML) can also provide significant help in the agricultural sector. The application of AI and ML for plant disease detection is being proposed due to the tedious, labor-intensive and time-consuming nature of manual plant disease monitoring (Nandhini *et al.*, 2019).



However, existing approaches for tomato leaf disease classification suffer from limited robustness and accuracy due to the complex and dynamic nature of the disease symptoms (Patel *et al.*, 2022). Therefore, there is a need for a more accurate system for classifying five tomato leaf diseases that can effectively distinguish between similar symptoms. This study proposes a deep generative adversarial network-based system for tomato leaf disease classification.

## 2.0 Literature Review

Machine learning algorithms, particularly deep neural networks, have shown promising results in tomato disease classification. However, these models are vulnerable to adversarial attacks that can cause misclassification of images. GAN can be used to increase the robustness of DNN against such attacks. This literature review aims to provide an overview of the use of GANs for tomato leaf disease classification. A tomato leaf detection system using GAN can automate the tracking of health and unhealthy tomatoes on the farm which will lead to solving one of the major global problems that humanity faces today which is food insecurity of which plant diseases are a major cause (Fazari, 2019). With the wide application of deep learning, many researchers focus on the identification of diseases and pests. In agriculture, the use of large datasets shows promising results for plant pathology with deep neural networks and ML to study the problem of disease identification (Hammou *et al.*, 2022).

Several research studies have delved into the application of deep learning techniques for plant disease classification, showcasing promising results. Gadade *et al.* (2021) utilized deep neural networks to achieve accurate disease diagnosis in different crops. Gadade *et al.* (2021) also proposed a transfer learning-based approach to address limited data challenges in plant disease datasets. Haque and Zisserman, (2022) used AlexNet and SqueezeNet models to classify and recognize five different types of tomato diseases in the Plant Village dataset. The experiment discovered that while AlexNet's classification accuracy is slightly higher than SqueezeNet's, the size of the model and the time required are doubled. Sharma *et al.*, (2018) used AlexNet and VGG16 in conjunction with transfer learning to identify seven types of tomato diseases; the experiment revealed that the accuracies were 92.29% and 91.49% respectively. Although transfer learning can accelerate model convergence and improve recognition performance, it is constrained by the original network structure. Shruthi *et al.*, (2022) proposed an image processing for leaf spot recognition system, with the four stages such as image acquisition, image segmentation, feature extraction, and classification. To compute the disease features, they also utilized the K-means algorithm. They achieved an accuracy of 90% and 80% for bacterial leaf spot and cotton leaf disease target spot, respectively. In their research, Gadade *et al.* (2022) introduced an end-to-end plant disease diagnostic model based on deep neural networks. This model offers reliable classification of plant types and plant diseases, closely simulating real-world user scenarios. The model comprises two main components: the leaf segmentation part, responsible for separating leaves from the background in the original image, and the plant disease classifier. The classifier employs a two-headed network that utilizes features extracted by multiple common pre-trained models to classify plant diseases. Experimental results demonstrated that this approach achieved an impressive plant classification accu-

racy of 0.8807% and a disease recognition accuracy of 0.8745%. Dandawate *et al.* (2019) proposed a leaf disease identification model based on generative adversarial networks (GANs) using instance normalization. This approach effectively recognized actual and false disease images on grape leaf lesions and utilized deep regret gradient penalty to stabilize training. The GAN-based data augmentation strategy successfully addressed overfitting in disease identification, leading to improved accuracy.

Zemouri *et al.* (2020) introduced a method using GANs to generate adversarial examples for cotton disease classification. They employed a DNN model to classify images of healthy and diseased cotton leaves and found that the model's accuracy decreased when exposed to adversarial examples. However, when trained with both original and adversarial examples, the DNN model's accuracy improved. Haque *et al.* (2022) proposed a framework combining GANs and DNNs for bean disease classification. They used GANs to generate synthetic images of diseased beans, which enhanced the performance of the DNN model for disease classification and reduced overfitting.

**Method and Technique:** The tomato diseases classification system using DGAN required a well-designed system architecture that will generate realistic images of tomato leaves (healthy and unhealthy) to solve the problem of low-quality images in training the model and also improve the accuracy result of the proposed model. The architecture of the system model is shown in Figure 1.

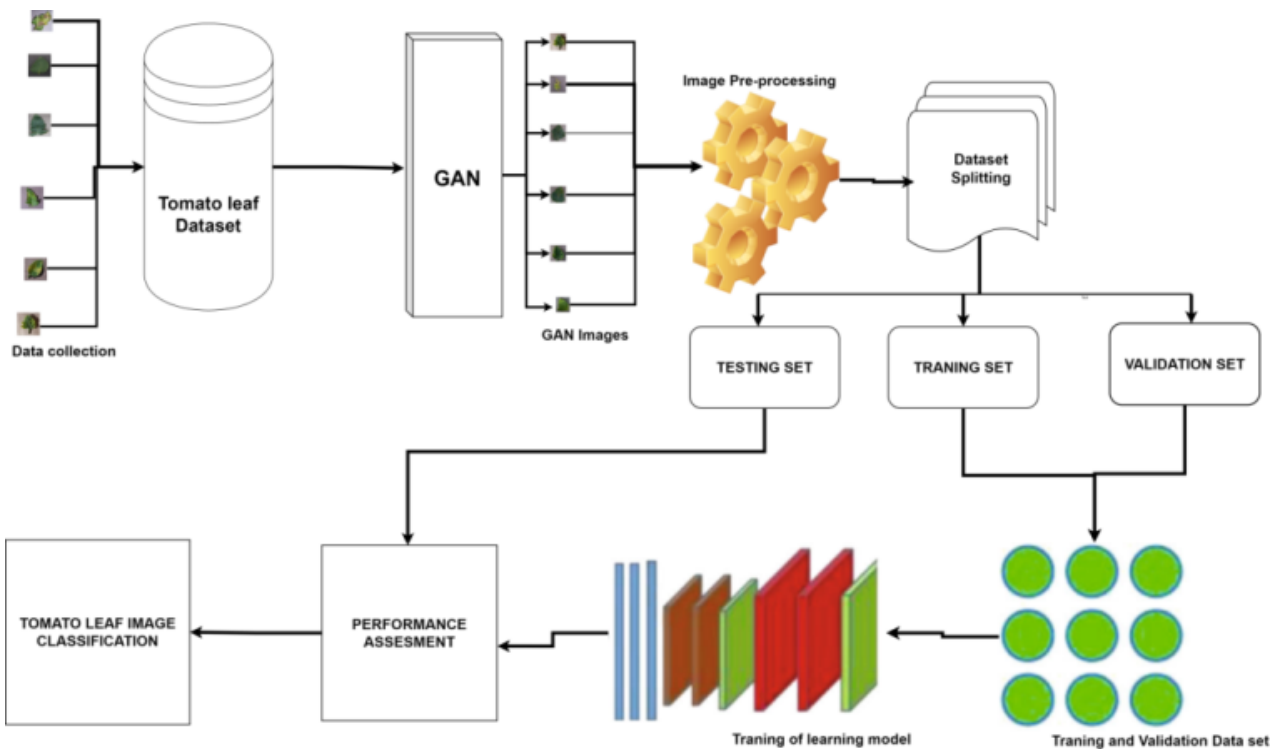


Fig 1: Architecture of the System Model

The architecture of the system is divided into seven sections: Data collection, GAN Architecture, Image pre-processing, Dataset splitting, model creation, performance assessment and classification system.

**Data collection:** Data collection was the initial phase of the system, which involved acquiring and processing tomato leaf images. This process includes gathering a diverse range of tomato leaves including 1700 samples of healthy tomato leaves and 1700 samples each for five different tomato leaf diseases including Tomato Leaf Spot, Early Blight, Late Blight, Bacterial Spot and Target Spot, which makes a total of 10200 instances of tomato leaves used for model training and validation.

Data collection was gathered from an online secondary source “Plant Village” which can be accessed and downloaded through the website <https://plantvillage.psu.edu/>.

**Deep Generative Adversarial Network:** Deep Generative Adversarial Network (DGAN) is a type of deep learning algorithm that uses machine learning and artificial intelligence to generate new data that is similar to the training data. DGAN is composed of a generator and a discriminator where the generator learns to create new data samples that resemble the training data. The generator network takes random noise vectors as input and generates synthetic data samples. The generator contains an encoder and decoder. The encoder is responsible for transforming input data from its original domain into a latent representation. This latent representation captures the essential features and information of the input data, which can be used to generate data in a different domain while the decoder takes the latent representation as input and generates data in the original domain. In other words, the decoder converts the latent representation back into meaningful and realistic data samples. Figure 2 shows an illustration of the DGAN structure.

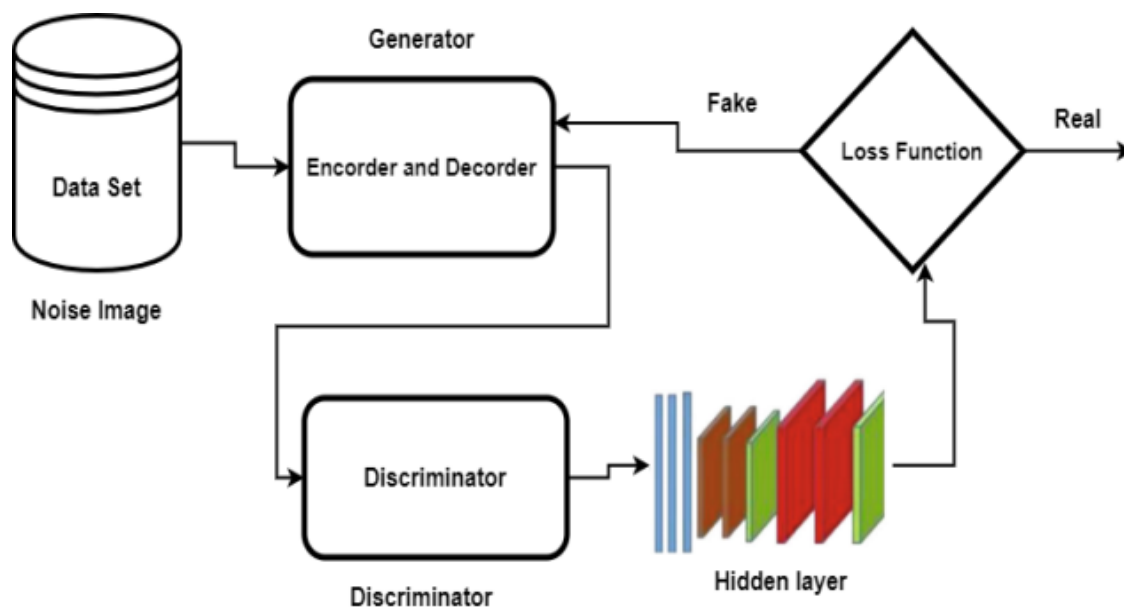


Fig 2: Deep Generative Adversarial Network

**Image pre-processing:** Image pre-processing is a deep feature extraction technique that was used to extract high-level representations of data from deep neural networks. This stage refers to the manipulation and analysis of the tomato leaf images using various algorithms and techniques which involves the application of mathematical operations, filters and extracts information from the tomato leaf images. In

image per processing resizing of images was also carried out, this step helps reduce computational overhead and ensures consistency during training also data balancing techniques were used to ensure that the model does not become biased towards the majority class. This involves oversampling the minority class or undersampling the majority class to achieve a balanced dataset.

**Dataset splitting:** Dataset splitting involves the distribution of the collected dataset into validation and testing datasets. The dataset was split using the 80:20 distribution where 80% is used for model training and 20% for model testing. Figure 3 shows a pie chart representation of the dataset distribution.

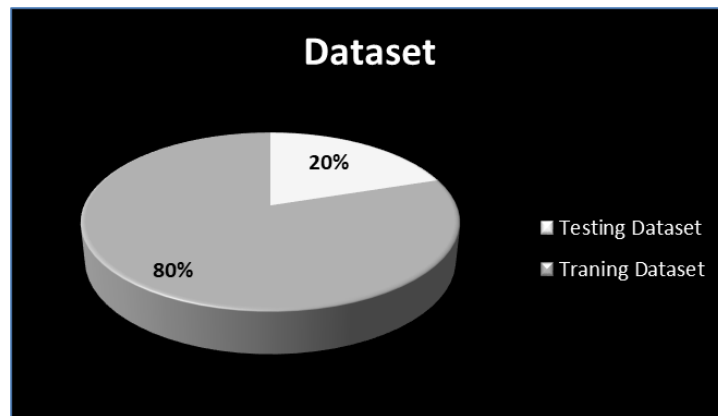


Figure: 3 Dataset distributions

Table 1 shows a breakdown of the dataset distribution of the various tomato leaf classes.

Table 1: Dataset distribution table

Tomato leaf	Healthy	Tomato Leaf Spot	Bacterial spot	Late blight	Early Blight	Target Spot	Total dataset
Training dataset	1,190	1,190	1,190	1,190	1,190	1,190	7,140
Testing dataset	510	510	510	510	510	510	3,060

### Model creation:

The model was defined as a sequential stack of layers using the `tf.keras.Sequential` class. This means that each layer follows the previous one in a linear order. The model started with two Conv2D layers, these layers perform convolutional operations on the input images to extract important features. The first Conv2D layer has 64 filters with a size of (3, 3) and uses the ReLU activation function to introduce non-linearity. After each convolutional layer, there is a MaxPooling2D layer with a (2, 2) pool size. Max pooling reduces the spatial dimensions of the feature maps, which helps to reduce the computational complexity of the model and makes it more robust to variations in the input data. Following the last max pooling layer, there is a flattened layer, which transforms the 2D feature maps into a 1D vector, preparing the data for the fully connected layers. The flattened layer is followed by two fully connected (dense) layers. The first dense layer has 256 neurons and uses the ReLU activation function. It serves as



a hidden layer that learns complex representations of the input features. The second dense layer has  $n\_classes$  neurons, corresponding to the number of output classes, and uses the softmax activation function. The softmax function converts the output scores into probabilities, indicating the likelihood of each class. Table 2 shows the model creation parameters.

Table 2: Model creation table

S/N	Layer (type)	Output shape	Parameters
1.	conv2d (Conv2D)	(None, 128, 128, 64 )	1792
2.	max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0
3.	conv2d_1 (Conv2D)	(None, 64, 64, 128)	73865
4.	max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 128)	0
5.	flatten (Flatten)	(None, 131072)	0
6.	dense (Dense)	(None, 256)	33554688
7.	Dense_1 (Dense)	(None, 64)	1542
8.	Total parameters: 33,632,878		
9.	Trainable parameters: 33,632,878		
10.	Non-trainable parameters: 0		

**Performance assessment:** The evaluation performance of the model was carried out using the following metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1_{score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

Where: TP = True positive

TF = True negative

FP = False positive

FN = False negative

**Classification system:** The classification system used in the model consists of two types of classification pattern which is binary and five-class classification pattern. The binary classification helps in identifying

healthy and unhealthy tomato leaves while the five-class classification system identifies the various tomato leaf diseases (Tomato Leaf Spot, Early Blight, Late Blight, Bacterial Spot, and Target Spot) under the unhealthy class of tomato leaves. Figure 4 shows the classification pattern used for the model.

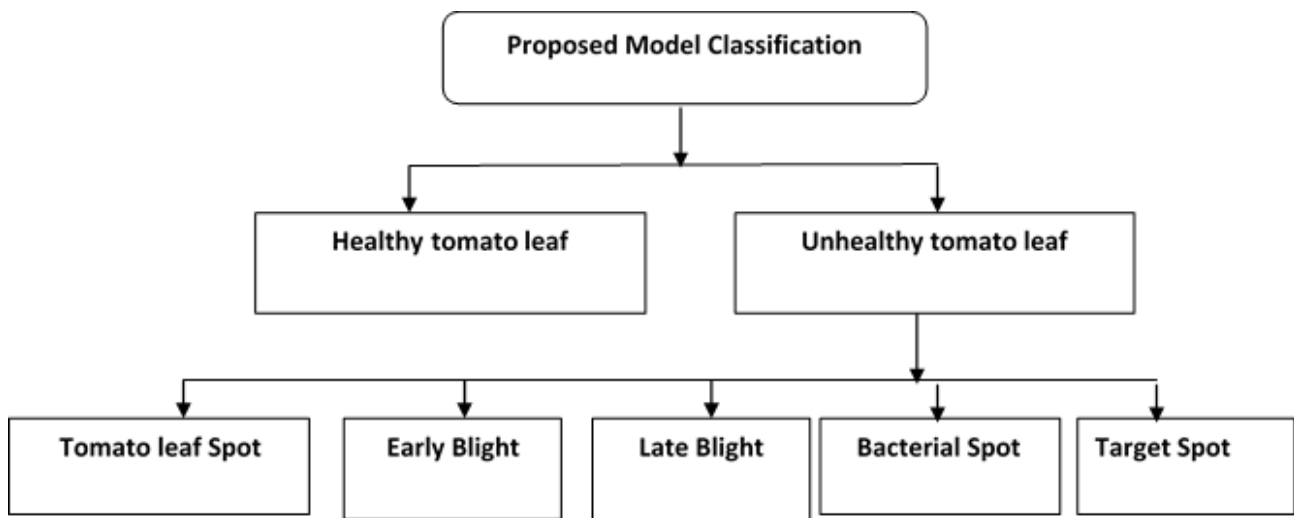


Fig 4: model classification system

**Experimental setup:** The system was implemented using a high-performance computer with a powerful graphic processing unit (GPU) (Nvidia GeForce GTX 1080 Ti or higher) with a sufficient RAM size of a minimum of 16GB and a large storage capacity of 1TB or higher. The software required a deep learning framework such as TensorFlow, Keras, or PyTorch to implement the model and OpenCV library for image processing and manipulation.

**Result and Discussion:** The classification report shows the Precision, Recall, F1-score and support values obtained for each class evaluated. According to the results, which are reflected in the confusion matrix, it was observed that the proposed model was able to predict the classes that were evaluated using the test dataset with 97% accuracy. Table 3 shows the classification report obtained in the evaluation of the proposed model.



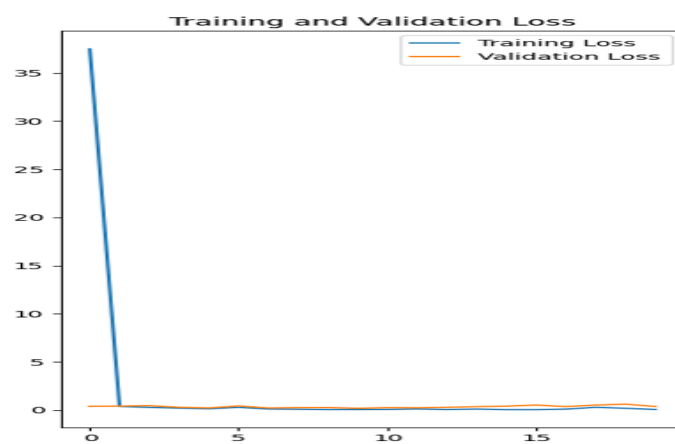


Table 3: Classification report

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
<b>Healthy</b>	0.96	0.96	0.96	227
<b>Tomato Leaf Spot</b>	0.97	0.92	0.94	181
<b>Early Blight</b>	0.95	0.95	0.95	115
<b>Late Blight</b>	0.98	0.99	0.99	332
<b>Bacterial Spot</b>	0.90	1.00	0.95	28
<b>Target Spot</b>	0.98	0.98	0.98	167
<b>Accuracy</b>		0.97	1050	
<b>Macro avg</b>	0.96	0.97	0.96	1050
<b>Weighted avg</b>	0.97	0.97	0.97	1050



Figure 5 shows the confusion matrix used to evaluate a model's performance by providing a detailed breakdown of its predictions for each class. It shows a more in-depth analysis of the model's accuracy in classifying different classes.



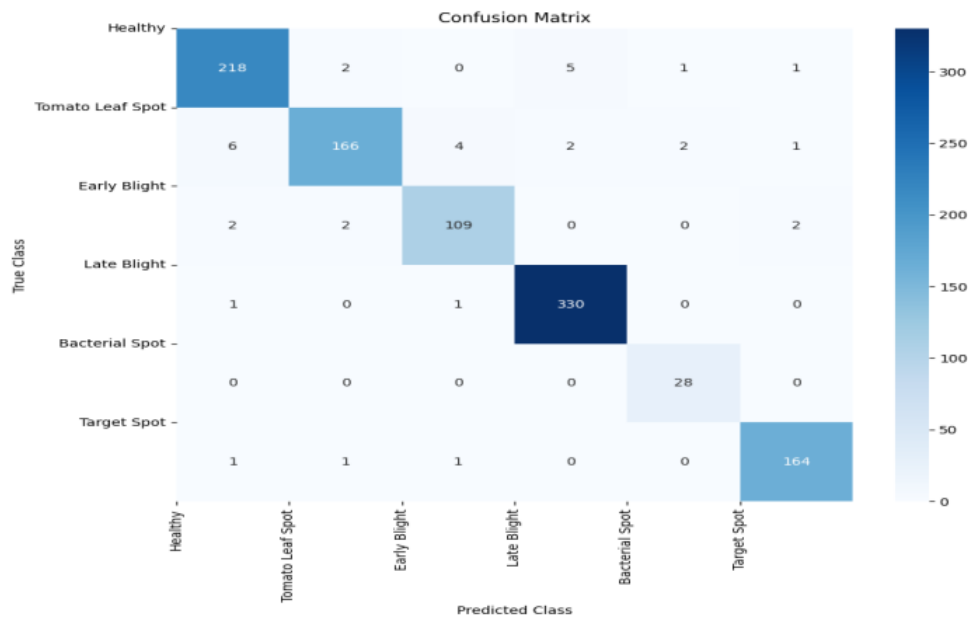


Fig 5: Confusion matrix

The rows represent the actual classes, the columns represent the predicted classes, and each cell in the matrix shows the number of instances belonging to a specific actual class that was predicted as a specific class by the model. In the first row of the confusion matrix, 218 instances of the "Healthy" class were correctly predicted as "Healthy," while 2 instances were incorrectly predicted as "Tomato Leaf Spot." In the second row, 166 instances of the "Tomato Leaf Spot" class were correctly predicted as "Tomato Leaf Spot," while 4 instances were incorrectly predicted as "Early Blight" and 2 instances were incorrectly predicted as "Late Blight." During the training process, a total of 1,015 images were successfully classified across 20 epochs.

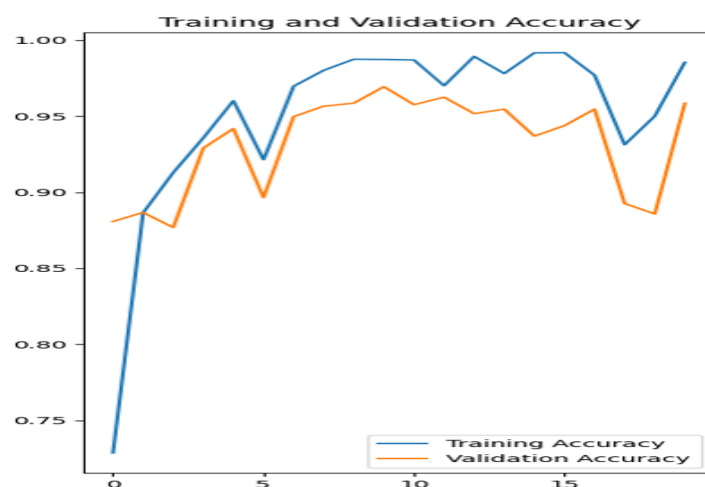


Fig 6: Accuracy results



The result shown in Figure 6 shows the overall performance of the classification model. It measures the proportion of correctly predicted instances out of the total instances. In this research, the model achieved an average accuracy of 97% during both the training and validation phases. It is noteworthy that the accuracy initially started at a lower level during the early stages of training. However, as the model was exposed to more data and training iterations, the accuracy gradually improved. This indicates that the model effectively learned from the data and refined its predictions over time. The fact that the model's accuracy remained consistently high on the validation set suggests that it has not suffered from overfitting.

Fig 7 Loss result

Figure 7 displays the loss function for both training and validation using categorical cross-entropy. Initially, the loss value was high due to random initialization of weights. As training progressed, the model improved, reducing the loss and making more accurate predictions.

**Model predictions:** The prediction process uses a dataset with a total of 1,350 images. The matplotlib library was used to visualize the prediction result. For each prediction, the image, the true result, and the result of the prediction made with the model were displayed, together with the percentage of accuracy. Figure 8 shows the results of the predictions made by the model.