International Journal of Social Science Archives

ISSN: 2707-8892

Available at www.ijssa.com

International Journal of Social Science Archives, July-September, 2024, 7(3), 820-833

Fungal Disease Identification of Tomato Leafs Using Deep Learning

Aamir Hussain[a], Jamshaid Iqbal Janjua[b*], Javid Ahmed Malik[c], Muhammad Saleem[d], Tahir Abbas[e*], Anaum Ihsan[f]

[a] Department of Computer Engineering, Modeling, Electronics and Systems, University of Calabria, Rende, Italy. [b,f]Al-Khawarizmi Institute of Computer Science (KICS), University of Engineering & Technology (UET), Lahore, Pakistan. [a,c] Department of Computer Science, Muhammad Nawaz Sharif Agriculture University Multan, Pakistan. [d]School of Computer Science, Minhaj University Lahore, Pakistan. [e]Department of Computer Science, TIMES Institute, Multan, Pakistan

[*]**Email:** jamshaid.janjua@kics.edu.pk, drtahirabbas@t.edu.pk

***Abstract*:** Tomato is a commercially significant crop, and manual identification of fungal diseases is inefficient and prone to human error, affecting export quality. We propose an automated method for detecting tomato fungal diseases using convolutional neural networks (CNN), addressing the gap in disease identification in Pakistan. To detect disease, a pre-trained CNN model was applied to a proprietary dataset of tomato leaf imagery. This approach efficiently identifies diseases like Early blight, Late blight, Septoria leaf spot, and Leaf mold. The automated method allows farmers to control these diseases, improving crop quality and boosting the country's agriculture economy through increased tomato exports. Our method offers a faster, more reliable solution than traditional manual identification.

Keywords: Tomato leaf, Fungal disease, Deep learning, CNN, Agriculture automation.

1. Introduction

Food security has become a worldwide concern, particularly in underdeveloped countries, where agriculture plays a crucial role in economic stability. In Pakistan, agriculture contributes around 19.8% to the GDP and enhancing vegetable production is essential for food security, poverty alleviation, and job creation [1]. Among vegetable crops, tomato plants are highly susceptible to variety of infections caused by fungi, bacteria, viruses, and nematodes, with fungal pathogens in particular, being the most common culprits behind yield reduction. Tomato plants are vulnerable to fungal diseases at every stage of their growth, which can be transmitted by seeds, water, air, soil, and vectors causing significant losses in production [2]. Food security in Pakistan continues to be challenged by the prevalence of plant diseases, particularly in tomato crops, a vital vegetable for the local diet. Recent advancements in crop management practices and biological controls are being explored to mitigate the effects of fungal pathogens in tomatoes. Tomato wilt, caused by pathogens such as *Fusarium oxysporum*, *Acremonium strictum*, and *Pseudomonas solanacearum*, is a major issue in Punjab [3]. The disease leads to symptoms like drooped leaves and stunted growth, with incidence rates between 20% and 33% in surveyed areas. However, the rapid evolution of these pathogens, combined with a lack of access to modern farming technologies in rural areas, hampers efforts to enhance production and ensure sustainable agriculture. Innovative strategies centered on cultivating disease-resistant varieties and employing integrated pest management systems are essential for ensuring food security and enhancing farmers' livelihoods in Pakistan. Effective disease identification and control are equally crucial for boosting crop yields and maintaining production quality.

Flow of Research

The hierarchy of this research begins with identifying Justification for an automated system to identify fungal illnesses in tomato plants, followed by the implementation of image processing methodologies and exploration of deep learning models. These tools are employed to recognize the disease from leaf images and suggest remedies for its treatment. This flow ensures an efficient approach to tackling the problem at hand.

Objectives and Novelty Statement

The purpose of this research is to create an automated disease identification system for tomato plants that employ convolutional neural networks. This study's originality comes in its execution of a custom-built dataset of tomato leaf images and the application of pre-trained CNN models to identify multiple fungal diseases, offering a faster, more accurate, and scalable solution for disease management in agriculture.

2.Related Literature

Recent studies have highlighted the complexity of using digital and smartphone camera images for plant disease detection. Challenges arise due to varying image qualities and environmental conditions, making traditional methods of disease recognition dependent on the expertise of farmers and specialists, leading to potential errors and misdiagnosis [4]. As deep learning and artificial intelligence have developed, automated methods like convolutional neural networks (CNNs) have become more efficient in diagnosing plant diseases, particularly fungal infections in tomatoes. These technologies reduce human error and enhance diagnostic accuracy by using image-processing techniques [5]. Researchers have utilized pre-trained CNN models like AlexNet, VGG16, and MobileNet to improve the efficiency of disease detection, achieving accuracy rates between 95% to 99% in various crops, including tomatoes [6]. The deployment of CNN-based models not only ensures early disease detection but also helps mitigate the economic impact by improving yield through timely disease management [7]. Moreover, explainable AI techniques, such as Saliency maps and Grad-CAM, have been employed to interpret these CNN models, ensuring that the AI’s decision-making process can be trusted by farmers and experts alike. This transparency enhances the usability of AI tools in real-world agricultural settings [8]. Automated systems like these, integrated into smartphone apps or other digital platforms, are proving vital in the precise identification and timely management of diseases, reducing losses and enhancing agricultural productivity. This paper explores how deep learning techniques are revolutionizing agricultural practices by enabling precise monitoring and diagnosis of plant diseases for improved crop health [16]. This study focuses on creating scalable infrastructures for handling the vast data generated by IoT devices, which can support agricultural applications by managing large-scale sensor data efficiently [17]. The work in [18], proposes a deep learning framework for detecting anomalies in medical images, providing insights into the use of similar techniques in identifying plant diseases through image analysis. The paper discusses IoT and big data analytics for improving healthcare delivery, which parallels the potential for IoT-driven solutions in enhancing agricultural disease monitoring [19]. This paper introduces an authentication system leveraging blockchain technology, offering a model for securing data in IoT-based plant disease identification systems [20].

Limitations

Despite advancements in convolutional neural networks (CNNs), challenges related to varying image qualities and environmental conditions persist, which can impact the reliability of disease detection systems [9]. Additionally, the performance of pre-trained CNN models may vary across different plant species and disease conditions, affecting overall diagnostic accuracy [10]. Moreover, while explainable AI techniques such as Grad-CAM enhance transparency, they may still fall short in providing comprehensive interpretability for complex models in diverse agricultural environments [11].

3.Methodology

In this study, the VGG16 v2 architecture was used. VGG16 detects many items in an image with a single shot. Even though VGG16 utilizes bounding boxes like Faster RCNN and the usage of these architectures is nearly identical, VGG16 performs well on mobile devices (Fig. 1). According to the volume of the entity to be identified, the bounding boxes have varying sizes and aspect ratios. Examples include human beings (bounding boxes can be rectangles), computer mice (bounding boxes can be squares), various flora leaf sizes, and so on. Every box has its class confidence score, representing the possibility that the recognized object is correct. Because models trained with VGG16 and the MobileNet feature extractor have higher mean average precision (MAP) values, VGG16 is expected to be faster than Faster RCNN (MAP), or more accurate than models trained with regional proposal network (RPN)-based techniques, of which Faster RCNN is an example. Despite this, the findings of the VGG16 training yield appear to be quite accurate when used with our trained detector. As a result of its excellent accuracy and compatibility with mobile devices, we chose SSD for our project. SSD utilizes VGG16 to obtain feature maps. Then, objects are detected using the Conv4 3 layer [12]. After VGG16, it adds six more auxiliary convolution layers, totaling 8732 predictions. The VGG model is a deep convolutional neural network developed by the University of Oxford scientists A. Zisserman and K. Simonyan [13] comprising about 138 million parameters. On the ImageNet dataset, this model achieved a 7.4 percent error rate. It increased AlexNet's performance [14]. Instead of 11x11 and 5x5 filters in the first two layers, the network created numerous smaller 3x3 filters one after the other by modifying the kernel size.

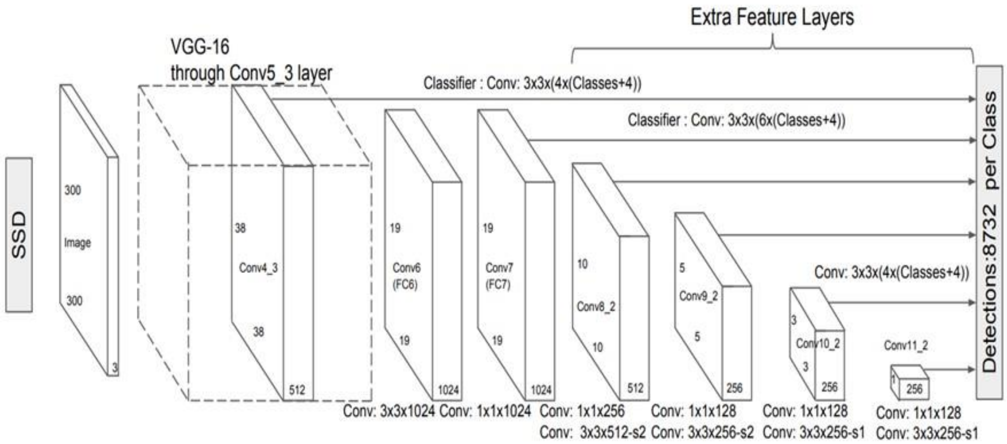


Figure 1: vg16V2 Architecture

SSD Grid-Based Detection

Alternative to utilizing sliding windows, the SSD method divides the image into grid cells, each of which is in charge of identifying objects in its assigned region. For every detection task, the class and location of the object within the cell's zone are predicted (Fig. 2). The grid cell sets a background class and disregards positional data in the absence of an item. For example, the system displays the shape and position of any object discovered within each grid cell using a 4x4 grid. You might be wondering what happens if there are multiple objects in a single grid cell or if we need to recognize a variety of objects. The anchor box and receptive field are used in this situation.

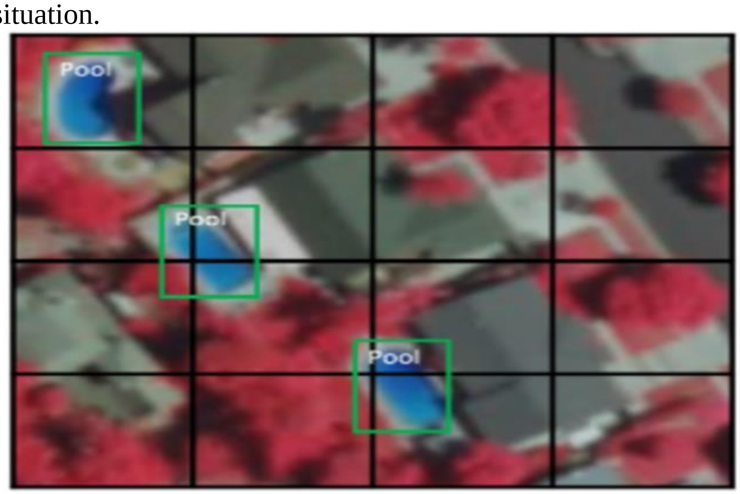


Figure 2: Example of a 4x4 grid

Anchor box

In SSD, several anchor boxes can be linked to each grid cell, with each anchor box predefined to control the size and shape. Taller objects are assigned to taller anchor boxes, while wider structures use wider ones. During training, a matching process connects anchor boxes to the correct bounding boxes in the image. The box with the highest overlap predicts the object's class and position. This method assists in training the network and in object location prediction. Aspect ratios and scales are assigned to each anchor box (see Fig. 3 for reference).

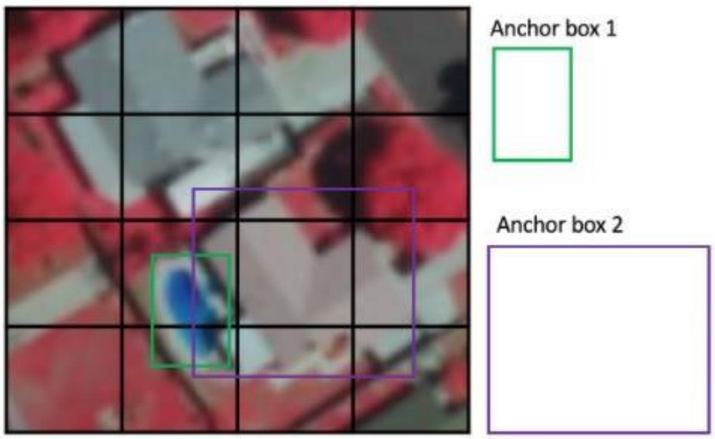


Figure 3: Example of two anchor boxes

Aspect ratio

Not everything has a square shape. Some are longer and some are wider to varying degrees. The SSD architecture allows for pre-defined aspect ratios of the anchor boxes, which makes this possible. The ratios parameter can be used to describe multiple aspect ratios of the anchor boxes associated with each grid cell at each zoom/scale level.

Zoom Level

Since anchor boxes are used to identify objects of different sizes, they do not have to match the size of grid cells. Users can scale the anchor boxes up or down, according to each grid cell's guidelines, to find things of all sizes. For example, anchor boxes have ratios different than swimming pools since structures are typically larger.

SSD

When a color image is input into the system, SSD performs the following steps:
Step 1: To generate feature maps, the image is processed through several convolutional layers at various phases.
Step 2: A 4x4 filter is used to determine a small default box in each of those feature maps.
Step 3: The bounding box offset is predicted for each box.
Step 4: The probability of each box's class is predicted.
Step 5: Using IOU, the predicted boxes are compared to the ground truth boxes.
Step 6: Instead of considering all negative cases, the best-assured loss for each default box is used in the final result. The steps in our proposed methodology are shown in Fig. 4.

Dataset Collection

Images acquired using a smartphone with a resolution of 13 Megapixels were used to create a dataset. These photographs were taken at three different farms. To reduce the loss of disease characteristics due to specular reflection. The effectiveness of deep learning models heavily depends on the amount of data used. We created a dataset from these edited images, which included five categories: four types of diseases and a healthy tomato leaf. To fit the input requirements of the models, the images were resized to 300 x 300 x 3 pixels. To focus on the disease-specific features and avoid interference from background information, we manually segmented the images using Adobe software to remove any background elements. Insufficient input data leads to overfitting of the learned model, which performs poorly during validation. According to the research, estimating the number of photos required for classification and identification is a challenge.

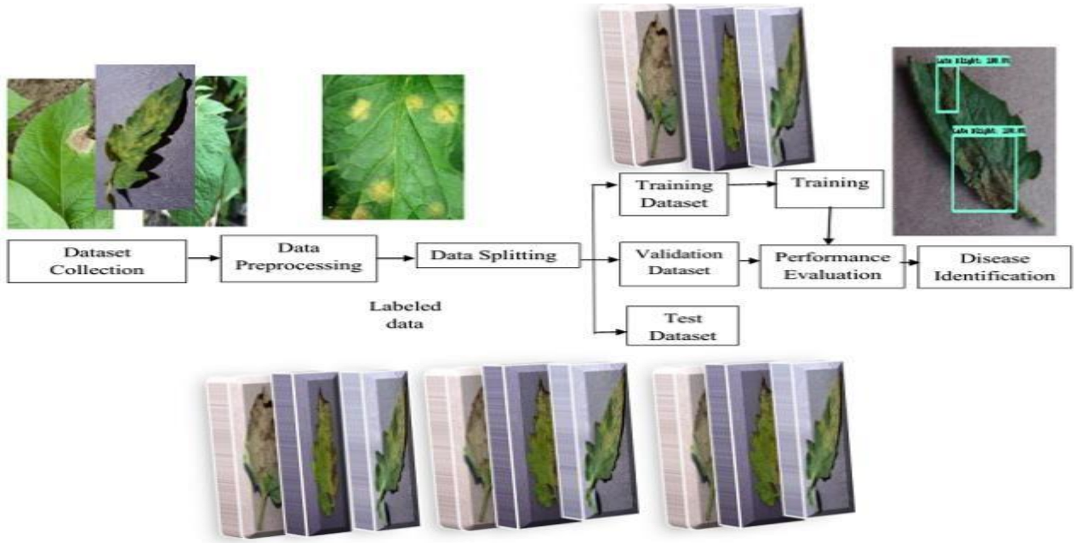


Figure 4: Methodology for determining best-assured loss for each default box

To conduct this experiment, a new dataset must be created. This involves gathering photos of tomato leaves from different tomato farms and from the Plant Village website. Images of diseased and healthy tomato leaf plants are included in the dataset. We created this system by extracting dataset from the well-known Plant Village collection was put together by mixing new photos of five different tomato leaf diseases, which we took with a digital camera. A healthy leaf, our dataset includes samples for five different forms of tomato plant disease with five classes in total, as follows: 1, Early Blight.2, Late Blight. 3, Leaf mold. 4, Septoria Leaf spot. 5, Healthy Leaf (Table 1).

Table 1: Distribution of Tomato images after splitting

Class	**Training Set**	**Testing Set**	**Validation Set**
Early Blight	800	100	100
Late Blight	800	100	100
Healthy Leaf	800	100	100
Leaf Mold	800	100	100
Sepotaria Leaf Spot	800	100	100
Total	4000	500	500

Image Preprocessing

The RGB images in our collected dataset and the Plant village databases are of any size. To guarantee that all important details for feature extraction are captured, each shot was manually cropped during processing. The photos were reduced to 300x300 pixels to accommodate the square format required by most neural network models while retaining a consistent aspect ratio. Background noise was reduced, and the intensity was adjusted for correct normalization.

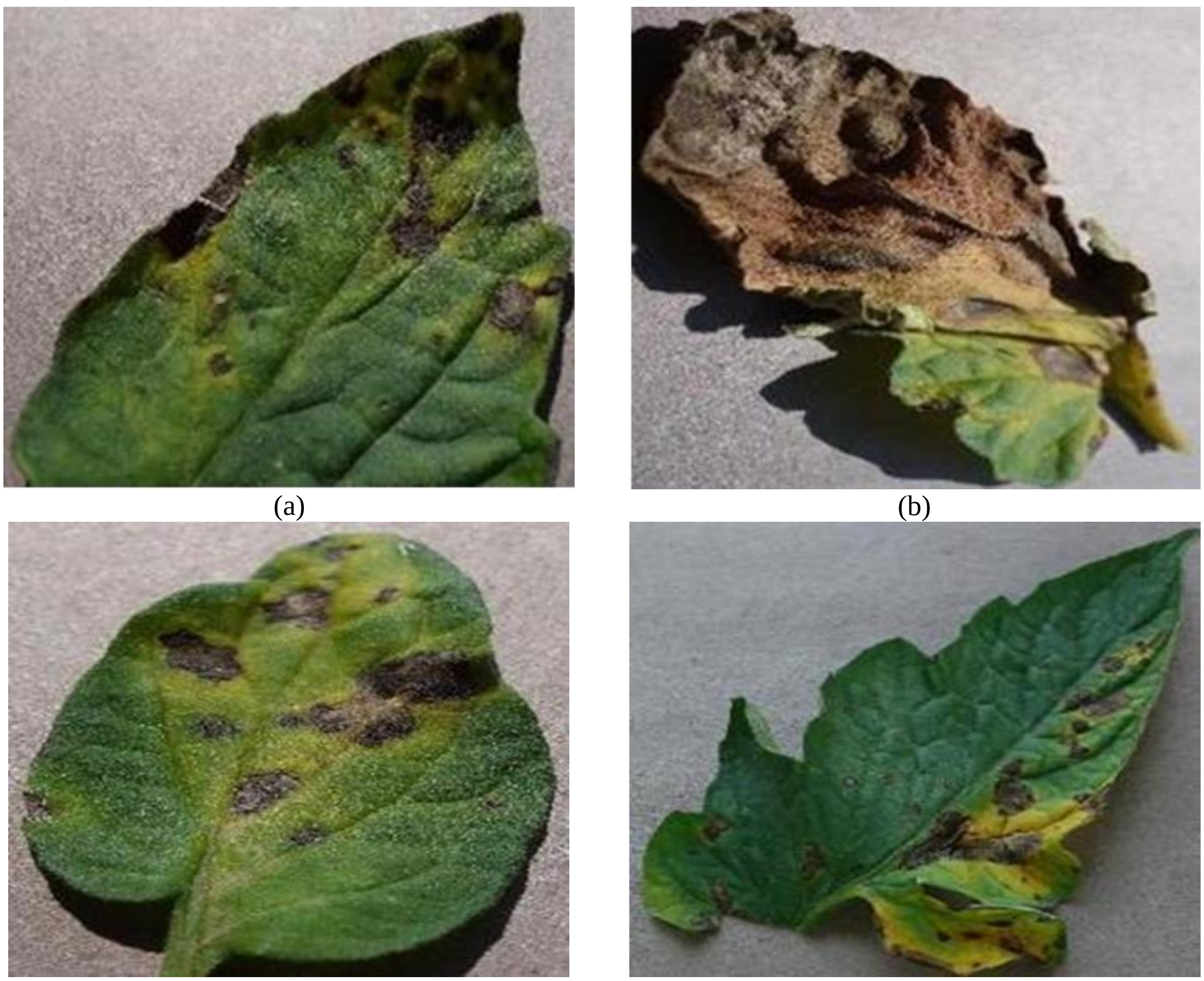
Boosting Models with Data Augmentation

By creating other copies of already-existing images, augmenting data helps address the problem of small datasets by enhancing model generalization. Research shows that even while these modified photos may appear redundant and are not as useful as fresh data, they still have hugely beneficial effects on improving model performance.

All the images in the dataset have been resized to 300x300 pixels to improve uniformity. Rotations ranging from 0 to 25 degrees were applied at random, and brightness was altered from 0.2 to 1.5. Random variations in the zoom levels were applied, along with small adjustments to the width and height. Flips that are both horizontal and vertical were added for more variation. To avoid having too dark or dazzling of an image, minor modifications were made to the RGB colors, staying within a 20% range. Data augmentation helps by expanding the dataset and also lets you see key features of images at different scales and positions. To prevent overfitting with the limited dataset, random data augmentation was used on the tomato leaf photos. Images were resized to 224×224 to ensure they were ready for consistent network input before training. In this study, different data augmentation techniques are used on training samples help to resolve over-fitting issues. The zoom refers to zoom-in/magnifying or zoom-out/reducing the picture built on arbitrary value between 1 to ±0.1. The rotation is termed as a spin angle in degrees like 10 degrees used to generate random pictures between -10 to +10.

Dataset Samples

Five thousand snapshots depicting common tomato leaf diseases, such as Sepotaria leaf spot, Early Blight, Late Blight, and Healthy Leaves, were included in the dataset. These were chosen among 1,090 real-time images that were shot in various lighting, temperature, and humidity situations. With 4,000 photos utilized for training, 500 for validation, and 500 for testing, the images were split into three groups for training, validation, and testing in an 80:10:10 ratio. Dataset samples Figure 5 class 1 represents the early blight disease of tomato in Fig (a) presents initially, small dark spots form on older foliage near the ground. Fig (b) and (c) present larger spots that have target-like concentric rings. Fig (d) shows the Dead, dry leaves may cling to the stem or highly infected foliage may turn brown and fall off.



(a) (b) (c) (d)

Figure 5: Class 1- Early Blight


(a) (b)


(c) (d)

Figure 6: Class-2 presents the healthy tomato leaf.

Fig 7 class 3 represent the late blight a lethal tomato disease may quickly spread across fields and, if left untreated, completely eat away a crop.


(a) (b)

(c) (d)

Figure 7: Class-3 Late Blight

Tomato leaf mold typically occurs in greenhouses with humidity levels of more than 85%. It mostly affects the leaves, causing them to wilt and potentially reducing total production as shown in Figure 8 class 4.

(a) (b)

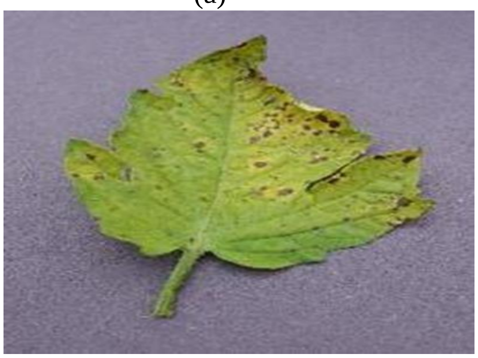
(c) (d)

Figure 8: Class-4 Leaf Mold

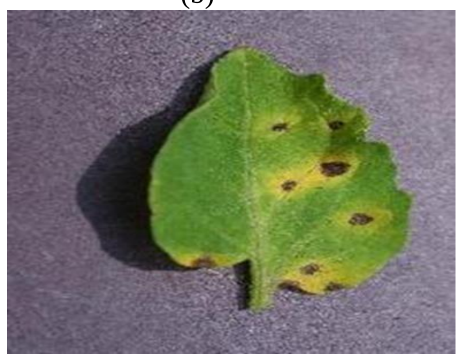
Septoria leaf spot, caused by the fungus Septoria lycopersici, shows up on the plants' older, lower leaves and stems when they develop fruit. However, it can also affect the petioles, stems, and calyx as shown in Figure 9 Class 5.

(a)

(b)



(c)



(d)

Figure 9: Class-5 Septoria Leaf Spot

Feature Extraction

After preprocessing, the image's illness component is retrieved. In image processing, leaf disease patches are initially detected as intersection zones. From there, additional features relating to the symptoms are obtained to aid in disease classification.

Disease Identification

The entire dataset was first separated into three sections: training, validation, and testing. By randomly splitting the 5000 photos, the training set became 80% and the test and validation set became 20%. By retraining our models, we were able to improve their accuracy through the application of transfer learning. Despite being distinct from the testing set, the validation set is essential for choosing parameters and avoiding overfitting during the training phase.

Model Training and Evaluation Process

The results and experiments of applying the VG16, SSDv2, and YOLOv5 algorithms to identify fungal infections in tomato leaves are described in this chapter. All testing details with experiment's results, plus a discussion of the findings, are clearly reported. The findings of the experiments are presented through various graphs and tables. The data sample used for training the model allowed for accurate classification of the test samples into appropriate classes. The test data, which contained new, previously unseen samples, was used to assess how well the model could correctly classify them. A model is considered underfit when it poorly classifies the test data due to not capturing the complexity of the training sample. Conversely, overfitting occurs when a model learns the training sample too closely, hindering its ability to correctly classify new test data. An epoch represents one complete pass through the training dataset; for example, setting the epoch to 50 means the model trains 50 times through the entire dataset, with both forward and backward propagation completed during each epoch. Generally, longer epochs improve model training and data prediction. Batch size refers to the set of samples used in one iteration of training. For instance, if there are 80 samples and the batch size is set to 16, the dataset is divided into

4 batches. The learning rate, a crucial hyperparameter, determines the extent to which network weights are altered based on the loss gradient, with even minor differences significantly affecting how models learn. Training time refers to the time taken by the model to complete the training process. Accuracy measures the performance of the classification during training, representing the proportion of correctly classified instances, while loss is measured to evaluate how well the model has learned to predict the correct class.

Image Categorization and Experimental Phases

The color characteristic was employed to categorize the image for the primary reason because when we look at the image, we see it in a variety of colors. Simply, the image displays individuals who are healthy, ill, or unhealthy. Our experiments are divided into two sections. The process is broken down into two stages: training, in which the classifier learns from data, and testing, in which performance is evaluated using unseen data. The next sections will go over the results of the experiment in detail.

Changing the Dataset Ratio for Training and Testing

The following table displays classification accuracy percentages for training, validation, and test data, based on the various split ratios utilized in the model studies.

Table 2: Experiment results employing various training and testing data ratios

Training RATION	**TEST RATION**	**ACCURACY**			**LOSS**		
		Training	**Test**	**Validation**	**Training**	**Test**	**Validation**
70	30	82.3%	83.1%	82%	14.2%	7.2%	10.1%
80	20	89.99%	89.99%	89.8%	4.3%	6.25%	5.7%
60	40	75.3%	76.6%	75.1%	20.5%	22%	18.2%

As shown in Table 2, the model produces promising results with various training and testing dataset ratios. In those trials, a superior outcome was obtained by allocating 80% for training, 10% for validation, and 10% for testing. than using the other two methods. When the 8:2 ratio is employed, it signifies that 80 percent of the dataset is employed in training and 10% is used for testing. In addition, ten percent of the entire dataset was validated.

Using Different Activation Functions

The training, validation, and testing data accuracy metrics are expressed as percentages and independently are shown in the following table based on the results of the model's experiment with different activation functions.

Table 3**:** The model's output with activation functions

Activation Functions	**ACCURACY**		**LOSS**	
	Training	**Test**	**Training**	**Test**
Soft-Max	89.99%	89.4%	0.0004%	0.006%
Sigmoid	85.6%	83.1%	15.2%	8.6%

As shown in Table 3 The output layer's SoftMax activation mechanism operates more effectively to the sigmoid for categorical classification. Finally, The model has an average test accuracy of 89.4% and a training accuracy of 89.99% while employing a learning rate of 0.0001, a SoftMax activation function, and an 80:20 train-test split.

Training with Original Dataset

Fig 10 shows that when training with the original dataset model achieves an accuracy of 75.23% and the loss is 6.08. This is because the dataset without preprocessing has noise.

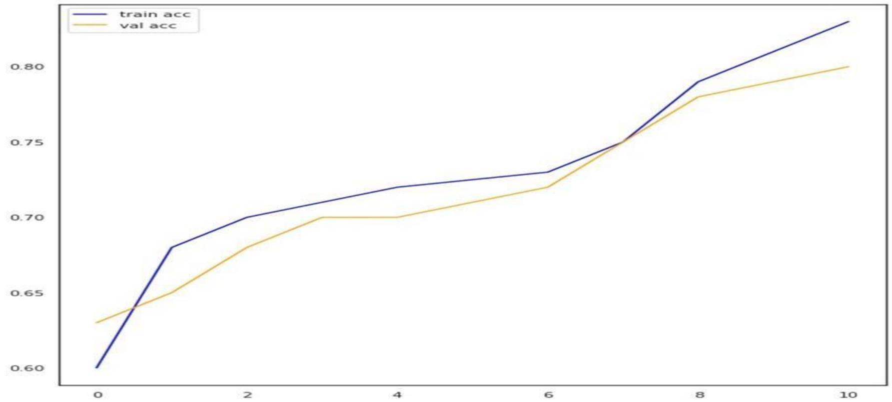


Figure 10: Training with the original dataset

Training with Preprocessed Dataset

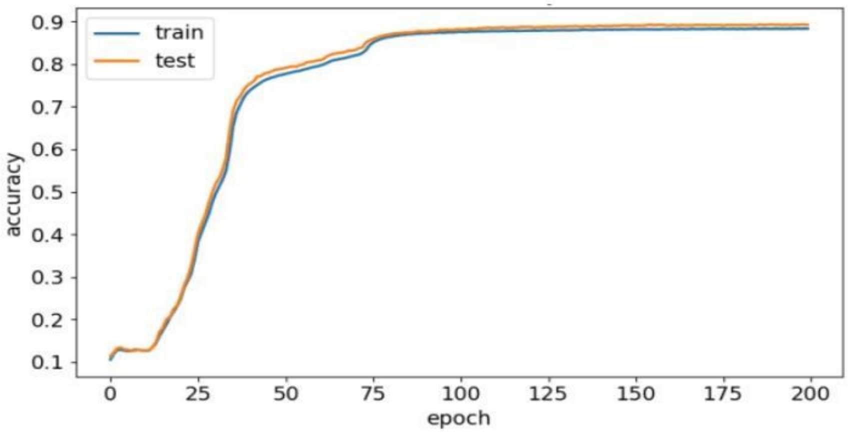


Figure 11: Training with Preprocessed Dataset

4.Results and Discussion

The trained model's picture findings are shown in Fig 12, Fig. 3.15, and Fig. 3.16. When we validated our trained model, we acquired the disease name and confidence scores. Table 3 presents the evaluation accuracy of five different tomatoes

Figure 12: Early Blight Identification

Figure 13: Leaf Mold Identification

Figure 14: Late Blight Identification

Table 4: Evaluation of classification model for fungal diseases in tomato leaves

Disease Type	Per Class Accuracy	Test Loss%	Learning Rate
Early Blight	92%	0.2	0.0001
Healthy Leaf	94%	0.1	0.0001
Late Blight	87.9%	0.5	0.0001
Leaf Mold	89.2%	0.2	0.0001
Sepotaria Leaf Spot	85.7%	0.18	0.0001

The Table.4 summarizes the classification model's performance in identifying fungal diseases in tomato leaves, showing per-class accuracy, test loss percentage, and a consistent learning rate. Each row reflects a different disease type, highlighting the model's accuracy and error rate for each category. The per-class accuracy shows the model's effectiveness in identifying diseases, with the highest accuracy for Healthy Leaf (94%), followed by Early Blight (92%) and Leaf Mold (89.2%). The model struggles slightly with Late Blight (87.9%) and Septoria Leaf Spot (85.7%). The test loss percentage, reflecting the model's error, is lowest for Healthy Leaf (0.1%), with Early Blight and Leaf Mold at 0.2%. Septoria Leaf Spot has a slightly lower error (0.18%) compared to Late Blight, which has the highest test loss (0.5%). The learning rate is consistent across all disease types at 0.0001, ensuring steady weight adjustments for balanced model performance.

5.Conclusion

Tomatoes are a popular fruiting vegetable grown worldwide. To meet global demand, it is essential to improve crop yield and ensure early identification of pests, and bacterial, and viral infections. Previously, disease diagnosis relied mainly on manual recognition approaches, such as assessing disease types based on farmers' experience or consulting agricultural information sources, taking images of disease specimens for internet searches, or consulting experts to examine disease characteristics. In this research, data was collected from five different sources, and pre-trained with an accuracy of 89.99%, the deep learning model recognized characteristics of prevalent illnesses in photos of tomato leaves.

References

[1] Adhikari, R., & Agrawal, R. K. (2012). Forecasting strong seasonal time series with artificial neural networks. Journal of Scientific & Industrial Research (India), 71, 657–666.

[2] Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Computer Science, 167, 293–301. https://doi.org/10.1016/j.procs.2020.03.234.

[3] Ali, S., Khan, J., Jadoon, A., Riaz, M., & Khan, A. (2020). Evaluation of farmer's socioeconomic characteristics influencing tomato output in District Peshawar, Khyber Pakhtunkhwa, Pakistan. Sarhad Journal of Agriculture, 36(2), 317–324. https://doi.org/10.17582/journal.sja/2020/36.2.317.324.

[4] Alam, M., Rehman, A., Iqbal, Z., Muhammad, S., Saira, M., Aslam, S., Bashir, M., Hameed, A., & Sarfraz, S. (2017). First Report of Tomato Wilt Caused by Plectosphaerella cucumerina in Punjab, Pakistan. Plant Disease, 101, 1543-1543. https://doi.org/10.1094/PDIS-01-17-0086-PDN.

[5] Cartolano, A., Cuzzocrea, A., & Pilato, G. (2024). Analyzing and assessing explainable AI models for smart agriculture environments. Multimedia Tools and Applications, 83(12), 37225–37246. https://doi.org/10.1007/s11042-023-14698-7.

[6] Bhargava, A., Shukla, A., Goswami, O., Alsharif, M. H., Uthansakul, P., & Uthansakul, M. (2024). Plant leaf disease detection, classification, and diagnosis using computer vision and artificial intelligence: A review. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3286018.

[7] Arge, F. O. R. L., & Mage, C. I. (2015). Vision-based detection and classification of leaf symptoms in rice. Journal of Science and Industrial Research (India), 74, 1–14.

[8] Azath, M., Zekiwos, M., & Bruck, A. (2021). Deep learning-based image processing for cotton leaf disease and pest diagnosis. In Proceedings of the 2021 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2021, 1–6. IEEE. https://doi.org/10.1109/COMITCon.2021.9418253.

[9] Kaundal, R., Kapoor, A. S., & Raghava, G. P. S. (2006). Machine learning techniques in disease forecasting: A case study on rice blast prediction. BMC Bioinformatics, 7, 1–16. https://doi.org/10.1186/1471-2105-7-485.

[10] Chhabra, A., Saroch, T., Gupta, R., & Shukla, A. (2023, November). Advancements in AI-assisted plant disease diagnosis. In 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 625–630). IEEE. https://doi.org/10.1109/ICCCIS.2023.10012045.

[11] Mahmud, T., Barua, K., Barua, A., Basnin, N., Das, S., Hossain, M. S., & Andersson, K. (2023, December). Explainable AI for tomato leaf disease detection: Insights into model interpretability. In 2023 26th International Conference on Computer and Information Technology (ICCIT) (pp. 1–6). IEEE. https://doi.org/10.1109/ICCIT54785.2023.10016025.

[12] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. Frontiers in Plant Science, 8, 1852. https://doi.org/10.3389/fpls.2017.01852.

[13] Dong, X., Wang, Q., Huang, Q., Ge, Q., Zhao, K., Wu, X., & Hao, G. (2023). PDDD-PreTrain: A series of commonly used pre-trained models support image-based plant disease diagnosis. Plant Phenomics, 5, 0054. https://doi.org/10.34133/plantphenomics.000054.

[14] Dey, P., Mahmud, T., Nahar, S. R., Hossain, M. S., & Andersson, K. (2024, January). Plant disease detection in precision agriculture: Deep learning approaches. In 2024 2nd International Conference on Intelligent Data Communication Technologies and the Internet of Things (IDCIoT) (pp. 661–667). IEEE. https://doi.org/10.1109/IDCIOT.2024.10012367.

[15] Rangarajan, A. K., & Purushothaman, R. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Computer Science, 133, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.105.

[16] Suri Babu Nuthalapati. (2022). Transforming Agriculture with Deep Learning Approaches to Plant Health Monitoring. Remittances Review, 7(1), pp.227-238.

[17] Aravind Nuthalapati. (2023). Building Scalable Data Lakes For Internet Of Things (IoT) Data Management. Educational Administration: Theory and Practice, 29(1), 412–424. https://doi.org/10.53555/kuey.v29i1.7323

[18] J. I. Janjua, T. A. Khan and M. Nadeem, "Chest X-Ray Anomalous Object Detection and Classification Framework for Medical Diagnosis," 2022 International Conference on Information Networking (ICOIN), Jeju-si, Korea, Republic of, 2022, pp. 158-163, doi: 10.1109/ICOIN53446.2022.9687110.

[19] Suri Babu Nuthalapati, & Aravind Nuthalapati. (2024). Transforming Healthcare Delivery via Iot-Driven Big Data Analytics in A Cloud-Based Platform. Journal of Population Therapeutics and Clinical Pharmacology, 31(6), 2559–2569.

[20] J. I. Janjua, M. Nadeem and Z. A. Khan, "Distributed Ledger Technology Based Immutable Authentication Credential System (D-IACS)," 2021 4th International Conference of Computer and Informatics Engineering (IC2IE), Depok, Indonesia, 2021, pp. 266-271, doi: 10.1109/IC2IE53219.2021.9649258.