Disease Detection in crops with Hybrid Architecture of ResNet and VGG16 Networks in Real Time

Manjunatha D
manjunath3divakar@gmail.com

JSS Science and Technology University

Gayathri S
JSS Science and Technology University

Irfan Ali
JSS Science and Technology University

Shanmukha S M
JSS Science and Technology University

Maseeulla Khan
JSS Science and Technology University

Article

Keywords: Machine Learning, Residual Network, Visual Geometry Group 16, Deep Learning, Disease Recognition

Posted Date: March 31st, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6194196/v1

Additional Declarations: No competing interests reported.

Abstract

Modern agriculture faces a critical challenge in ensuring the health and yield of crops, with the early detection and treatment of diseases playing a pivotal role. Traditional methods reliant on manual inspection are not only time-consuming but also prone to errors, leading to significant crop losses and economic impact. The importance of this research lies in mitigating these inefficiencies by developing a robust ML model capable of accurately identifying diseased tomato leaves in the plants. Datasets considered while algorithm training comprise around 6 thousand to 8 thousand images, including both healthy and diseased leaves. The primary gap this research addresses is the inadequacy of conventional methods in providing timely and accurate disease detection. The research aims to harness the power of advanced ML algorithms including Visual Geometry Group 16 & Residual Network for the deep feature extraction. The model is trained, validated, and tested using a split dataset approach to ensure high accuracy and reliability. Key findings indicate that the model achieves an accuracy of around 95% in simulations, although a slight drop in accuracy is observed in real-time applications. Despite this, the ML-based approach significantly surpasses traditional methods, offering a more efficient and scalable solution to find the diseased leaves soon in tomato. These findings highlight potential in ML to transform agricultural practices by providing timely and accurate disease detection, reducing the reliance on manual labor, and contributing to increased crop yields and sustainable farming practices.

Introduction

Backbone of the global economy is agriculture, and ensuring the health of crops is paramount for food security and the livelihoods of millions of farmers. Yet, a persistent problem is the diseases in leaves or plants, leading to significant crop losses each year. Traditional methods detect diseases based on human monitoring that is both prolonged and also leads to mistakes. Farmers struggle in identify diseases early, resulting in extensive damage and decreased yields.

In recent years, technological advancements in ML and IoT have evolved as new methods to address these challenges. By harnessing the power of these technologies, we can devise a system that detects the diseases precisely[1][2].

Many methods have been developed over time to recognize and identify these types of disease using various methods. They are discussed in this section.

A novel approach was proposed by [3] for the detection of diseased plant leaves using an autonomous robot and CNN. The robot was equipped with Arduino Uno for navigation and a smartphone camera. The camera does the live video stream to a laptop which does the image processing where CUDA and darknet were used for the processing part and the SORT algorithm was used for the tracking part of the video frame.

A model presented by [4] utilized a pre-trained ResNet 101, based on CNN, to process a tomato leaf dataset. The classification employed ECOC and achieved effectiveness of 98.8% with error 1.15%.

Projections were made based on a Support Vector Machine classifier.

An automated detection system introduced by [5] leveraged a backpropagation CNN (BPNN) algorithm to detect diseases at early stages, reducing crop damage. This system achieved 89% accuracy using BPNN, which was highest among the three BPNN, SVM, and KNN. The process involved converting leaf images to grayscale, applying noise filtering, segmenting images based on statistical features like standard deviation and skewness, and training classifiers including SVM, KNN, and BPNN and with the selected features.

Research by [6] focused on detecting diseased leaves using various techniques such as conversion of colour space and CNN segmentation methods such as morphological operations and region growing using ULCRF (Unsupervised Learning Conditional Random Field). Hyperspectral image analysis compared SDA (stepwise discriminant analysis) and SAM (spectral angle mapper).

A novel system by [7] [21] employed the You Only Look Once (YOLO) algorithm for real-time diseased plant detection, processing leaf images at 45 frames in a second. YOLO's bounding box prediction enhanced accuracy and used darknet and OpenCV for result views and training.

A new framework, PCA DeepNet, introduced by [8], combined PCA (Principal Component Analysis) with a designed DNN. This hybrid framework incorporated a GAN (Generative Adversarial Network) to enhance dataset quality. The system achieved a 98.5% accuracy and used a Faster Region-Based Convolutional Neural Network (F-RCNN) for disease spotting in leaves.

A study by [9] recognized diseased and undiseased corn leaves using pre-processed CNN models DenseNet121 and EfficientNetB0 for features extraction. Data augmentation addressed complex data cases. The model achieved a 98.56% accuracy, outperforming other pre-trained models like InceptionV3 and ResNet152 which required more processing power.

The modified Lermurs Optimization Algorithm, implemented by [10], served like filter-based characteristic transformation technique to enhance the accuracy of diseased paddy leaf detection. It extracted 7 box-cox statistical attributes from every thermal picture. Testing four classifiers, the K-Nearest Neighbour classifier achieved the highest accuracy of 90% using the proposed algorithm.

A modified deep learning model for disease detection in apple leaf in real time was proposed by [11], based on CNN and using Google Net Inception design along Rainbow concatenation. Training on a dataset of 26,377 images, the INAR-SSD model achieved 78.8% accuracy in early detection.

A CNN-based method by [12] employed 1D-CNN for feature extraction to identify diseased potato leaf spots. After calibrating data through rough and accurate calibration, the 1D CNN outperformed traditional SVM, achieving 97.72% accuracy with a significantly reduced average recognition time.

Research by [13] [19] utilized a robust DL ResNet 34 implemented Faster-RCNN for classification, incorporating a CBAM (Convolutional Block Attention Module) to extract deep key features. This model achieved a 98.1% accuracy.

A modified CNN model, LeNet, was proposed by [14] for detecting and identifying diseases of plant leaves, emphasizing minimal computing resources. It automatically extracted features for classification and achieved a 94% accuracy, outperforming AlexNet and GoogleNet. The system employed computer vision with DL alogorithms trained on public datasets containing both healthy and diseased leaves [15]. In a simulation study involving 15 cases, the model achieved an 88.8% accuracy, correctly identifying 12 diseased and 3 healthy leaves.

A hybrid approach by [17] [20] used VGG16 architecture and a algorithm to optimize weights in validation and testing, achieving a 95.9% accuracy on tomato leaves but with high validation loss. An alternative approach addressing overfitting achieved a 96% accuracy using an L2 regularization model.

A segmentation-based approach by [18] [22] automatically segmented diseased areas in diseased tomato leaves. Segmented regions were analyzed for disease categorization along with measuring severity using various models including KNN classifier, SVM classifier and LDA (Linear Regression Analysis) and Decision Tree classifiers on a dataset of 3,000 image datasets.

A comparison with various methods, algorithms used and their accuracy is shown in Table 1.

Table 1
Various Approaches by different studies

Method/Algorithm	**Datasets/Features**	**Accuracy**
C-YOLOX model, YOLOX-S model [23]	Tomato Diseased Leaves	84.42%
Convolutional Neural Network (CNN), Squeezenet architecture [24]	Healthy & Diseased leaves of tomato	86.92%
PCA (principal component analysis) and CV (coefficient of variation) [25]	Plants of sweet pepper	90%
PCA DeepNet, Generative Adversarial Network (GAN) [8]	Images of diseased potato leaves	98.55%
F-RCNN [19]	Leaf Miner Target Spot & Phoma Rot images of tomato leaves	91.67%
VGG16 & ResNet along with CNN (Proposed Method)	Around 8000 Images of Healthy and diseased Tomato leaves	94–96%

This research aims in leveraging these innovations to design reliable and working solution in monitoring the health of tomato plants

Algorithms and methodology

The methodology for the machine learning-based system for detecting diseased tomato leaves involves several key steps, encompassing data collection, model development, hardware integration, real-time detection, treatment application, and user interface. Figure 1 shows the overview of the block diagram.

Data Collection: Initial step in the software implementation involves the collection of a comprehensive dataset. This project utilizes open-source datasets from Kaggle, which include thousands of images depicting both healthy and diseased tomato leaves. The dataset spans multiple categories of diseases, ensuring a robust training process for the machine learning model. The images are diverse, representing various conditions under which the leaves might appear, including different lighting and environmental settings. This variety is crucial as it allows the model to generalize well across different scenarios it might encounter in real-world applications. The data collection phase ensures that the dataset is sufficiently large and representative, providing a solid foundation for the subsequent steps.

Data Pre-Processing: As Datasets are collected, they undergo extensive processing for enhancing its quality along with the utility for training the machine learning model. This phase includes augmentation, where methods such as scaling, color adjustments, and flipping are used to artificially expand the dataset and introduce variability. This helps in making the model more robust to different conditions. Following augmentation, dataset will be divided into three various sets. Training datasets will be used while training of the model, validation dataset helps in tuning hyperparameters and preventing the model overfitting, test datasets are considered for assessing the algorithms final results. The splitting makes sure that the model is evaluated on previously unseen images, providing an unbiased evaluation of its accuracy and generalization capability.

Designing the Model Architecture: Designing the model architecture is a critical step that involves selecting and configuring appropriate machine learning models for the task. For this project, two well-known architectures, VGG16 and ResNet, are employed. VGG16 is known to it’s effective and simple image categorization methods, using a deep network of 16 layers to extract features. ResNet, on the other hand, introduces residual learning, which helps in training very deep networks by addressing the vanishing gradient problem. These architectures are chosen for their proven track record in image recognition and their potential for finding the patterns in the images. The combination of these models ensures that the system can accurately identify and differentiate between healthy and diseased leaves.

Training of the Model and Optimization:

During the training phase, the pre-processed data is fed into the selected model architectures. The models are trained using the training dataset, where they learn to identify patterns and features that distinguish healthy leaves from diseased ones. To improve the training process, hyperparameters such as learning rate, quantity of batches, and the no. of epochs are properly changed. Dropout and regularization strategies are used to prevent overfitting and guarantee the model's performance with fresh data. Iterative adjustments to the algorithm’s values are made during the training process with the goal of minimizing the loss operation, that gauges the difference amongst the true labels and the projections made by the model. Continuous monitoring and adjustments are made based on the performance metrics observed on the validation set.

Testing and Evaluation of the Results: After training, algorithm's results are rigorously checked using test dataset, which was not seen by the model during training. This step involves generating various metrics parameters to determine how well the algorithm can classify new, unseen images. Confusion matrices are also analyzed to understand the types of errors the model makes and to identify any specific classes where it might be underperforming. Visualization tools like heat maps are used to inspect which parts of the images the model focuses on during classification, providing insights into its decision-making process. The results of this evaluation phase are crucial for understanding the model's strengths and limitations, guiding further refinement and optimization to improve its performance in real-world applications.

Various CNN architectures are used to design the disease detection machine learning module, those are explained in this section including VGG16(visual geometry group 16) and ResNet (Residual Network) architectures through which the features a = of the datasets are extracted.

CNN is an form of DL method build to process and analyze text input. In this process, the text is first converted into a matrix format. The initial step in using a CNN involves providing this text input. Next, the convolution operation is carried out within the convolution layer, a simple depiction of CNN is shown in Fig. 2.

A crucial component of CNNs is the ReLU. This function is a straightforward: the output is same if positive and zero if negative. ReLU is widely used in various neural networks because it simplifies the training process and often leads to better performance. Its primary role is to introduce non-linearity into the model. Another important aspect of CNNs includes Pooling, which serves to decrease dimensions.

This reduction helps to minimize computational load, making the processing more efficient. Essentially, pooling helps to downsize the image dimensions without losing significant information. It can be described using the following equation as an example.

$$\mathbf{X(y) = sign(2\sigma_{Sigmoid}(M_3\sigma_{relu}(M_2\sigma_{relu}(M_1y + n1) + n2) + n3) - 1} \qquad \text{-}\mathbf{(1)}$$

Here, y represents the input, and the bias of the length is represented by n. Each M and b represent a layer (first & second hidden layer and output layer) with a different number of neurons In two dimensions, the image convolution can be represented as below:

$$\mathbf{Y_{mn}} = \sum_{\mathbf{i=1}}^{\mathbf{l}} \sum_{\mathbf{j=1}}^{\mathbf{l}} \mathbf{K_{ij} x_{p(m,i),q(n,j)}}$$

-(2)

Where, K, Y, and x represent kernel, matrix, and the input image. p(m,i) and q(n,j) are index functions. The benefit of pooling is that it reduces dimension and computation, and it reduces overfitting tolerance towards small distortions.

Similarly, the VGG16 model is a CNN that is known for its depth and effectiveness. It contains a total of twenty-one layers. However, only 16 of these layers are Weight layers, meaning they have learnable parameters. VGG16 is specifically designed to be 16 layers deep. Figure 3 illustrates the representation of VGG16 architecture in a basic manner. This network has already undergone pre-training after being trained on over a million pictures from the ImageNet collection. This model has the ability to classify photos into a thousand distinct groups, covering a wide variety of objects like mice, keyboards, pencils, and various animals.

VGGG16 also consists of various layers which are CNN layers, layers of pooling, flat, and softmax. The convolution layers are similar to that discussed above. The max pooling layer finds the maximum value within the given window, that is:

$X_{pqr} = \max_{(p*t<=l<p*t+f),(q*t<=l<q*s+f)} a_{(l,x,r)}$ -(3)

where p, q, r represents height, width, and channel, and t & f represent the stride size. where the width of the window f is calculated using:

$f = S_L + 2W - t\ (S_{L+1} - 1)$ -(4)

where w represents padding.

Also, ResNet34 is a 34-layer CNN known for its advanced picture cactegorization capabilities. This model is pre-trained over the ImageNet data that includes over 100,000 pictures spanning 200 different categories. One of the key innovations of ResNet34 is the introduction of residual-block that help address difficulty in disappearing or high gradients that can occur in deep networks. These blocks utilize a technique called skip connections, where the activations of one layer are passed directly to a subsequent layer, bypassing one or more intermediate layers. This creates a residual block, and stacking these blocks forms the core of ResNet34. A depiction of ResNet architecture is shown in Fig. 4 along with the its basic element in Fig. 5.

The architecture of ResNet34 is inspired by the VGG-19 model but with an important modification: the addition of shortcut connections. These shortcuts transform the basic 34-layer network into a residual network, enhancing its performance and training efficiency. In essence, a residual network (ResNet) is a type of CNN where the fundamental component is the residual block, characterized by its skip connections. This innovative design is illustrated in Fig. 5 below:

The major software implementation is designing and training the ML algorithm. Datasets for training algorithms are taken from Kaggle open-source database. The trained model includes around 6000 to 8000 images including three types of diseased leaves of tomato plants and also healthy leaves. The datasets also include some other images designed to detect the environment when any tomato plant leaves are not shown to the camera. The list is shown as a snapshot in Fig. 6.

After linking the drive (Which contains the datasets) to Google Colab, then a path directory is created to access the images. Figure 7 shows some of the random images from the considered diseases have been printed on the colab notebook for visualization of the datasets considered.

The model is then divided into train, test, and validation datasets where validation dataset images are utilized to verify algorithm accuracy in the training phase. Algorithms use both VGG16(Visual Geometry Group 16) and ResNet(Residual Network) algorithms for feature extraction and training of the model. This ensures that the 20 features are extracted in depth to detect the diseases in the tomato plant leaves. The learning rate during the training phase can be observed in Fig. 8.

Results and Discussions

The software implementation phase involved training a machine learning model on datasets obtained from Kaggle, containing healthy and diseased pictures of tomato leaves. Leveraging VGG16 and ResNet algorithms facilitated thorough feature extraction, enabling accurate disease detection. After training the datasets through various algorithms and through three epochs, the accuracy of the simulation is obtained around 95% (Fig. 9).

The 4.1 shows various losses and accuracy in trained algorithm over training period. Summary for various diseases (Some other than the diseases considered) can also be seen with the accuracy for each of the disease in Fig. 10.

After the simulation, Fig. 11 depicts the confusion matrix, and the heat map is plotted to analyze the accuracy for each of the trained classes (Diseases in tomato leaves).

The heatmap generated after training illustrates the performance of the machine learning model across different classes. Each cell in the heatmap represents the model's accuracy for a specific combination of predicted and actual classes. This visualization (in Fig. 12) helps to identify any patterns of misclassification or areas where the model performs particularly well. Heatmaps become a valuable tool for model evaluation.

Following successful model training, it was deployed on a Raspberry Pi for real-time application, demonstrating its efficacy in practical agricultural settings. The accuracy plot is shown in the below Fig. 13 and Fig. 14.

Considering the confusion matrix, various parameters for the performance evaluation is calculated. Table 2 is a list of these various parameters for better understanding.

Table 2
Performance Evaluation using various parameters

Class/Metric	Precision	Recall	F1-Score	Overall Accuracy
Bacterial-spot	95.37	9600	9568	
Early-blight	88.31	9067	8947	
Healthy	99.33	9867	9900	
Late-blight	92.81	9404	9342	
Leaf-mold	97.93	9467	9627	
Overall accuracy				9493

The real-time detection is done using a camera and streamlit function to launch a website. Python code is used for the implementation of this function.

Conclusion

The model has successfully shown various uses of advanced ML algorithms in detecting diseases of tomato leaf. By leveraging a comprehensive dataset from Kaggle, which includes pictures of both diseased and un-diseases leaves of tomato, the model is meticulously trained to distinguish between different disease states with high accuracy. The training phase involved division of datasets to three different categories, ensuring robust evaluation and fine-tuning.

The use of sophisticated CNN methods (VGG16 & ResNet) was crucial for getting a simulation accuracy approximately 95%. These architectures are well-regarded in feature extraction of relevant parameters from images, which is essential for accurate disease classification. Throughout the training process, the learning rate and other hyperparameters were carefully adjusted to optimize algorithm's effectiveness, as visualized in training logs and plots.

In practical terms, the software demonstrated a significant capability to identify diseases in controlled conditions. However, the transition to real-time applications revealed a slight drop in accuracy. This discrepancy highlights the inherent challenges of real-world deployments, such as varying lighting conditions, different angles of leaf presentation, and potential occlusions. Despite these challenges, the real-time performance remains commendable, indicating algorithm's robustness along with effectiveness in training regime.

The outcomes of this software implementation are promising, suggesting that machine learning can be a valuable tool in agricultural disease management. The ability to recognize and identify the disease will provide insights to farmers take timely actions, potentially reducing crop losses and improving yields.

Furthermore, the insights gained from this project provide a solid foundation for future improvements, like use of more datasets and refined algorithms, and improving real-time processing capabilities.

Some of the main advantages of the model are:

- High Accuracy in Controlled Condition
- Utilization of Advanced Algorithms
- Real-Time Processing
- Automation

Some of the limitations are:

- Reduced Accuracy in Real-Time Applications
- Dependency on Quality of Input Data
- Complexity in Implementation
- Complexity in Implementation

In future, the machine learning model can be integrated with a hardware module for real-time analysis. The efficiency of the model can be enhanced using various and quality datasets to extract the features correctly

Declarations

Acknowledgements: The project is a scientific study in the field of agriculture and technology conducted by the Electronics and Communication Department in JSSSTU, Mysuru

Funding: Not Applicable.

Data availability: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of interest: We wish to confirm that there are no conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Code availability: The code is available on reasonable request to the corresponding author.

Ethics and Consent to Participate: not applicable

Consent to Publish: not applicable

References

1. C. Vengaiah and M. Priyadharshini,” CNN Model Suitability Analysis for Prediction of Tomato Leaf Diseases”, 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 1-4, doi: https://doi.org/10.1109/ISCON57294.2023.10111996.
2. R. Sharma, V. Kukreja and D. Bordoloi,” Deep Learning Meets Agriculture: A Faster RCNN Based Approach to pepper leaf blight disease Detection and Multi-Classification”, 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-5, doi: https://doi.org/10.1109/INCET57972.2023.10170692.
3. P. S. A. Kesava and K. P. Peeyush,” Autonomous Robot to Detect Diseased Leaves in Plants using Convolutional Neural Networks,” 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 806-809, doi: https://doi.org/10.1109/ICOEI.2019.8862737.
4. M. Kaur and R. Bhatia,” Development of An Improved Tomato Leaf Disease Detection and Classification Method,” IEEE Conference on Information and Communication Technology, Allahabad, India, 2019, pp. 1-5, doi: https://doi.org/10.1109/CICT48419.2019.9066230.
5. P. R. SAVARAPU, M. SUDHAKAR, Y. TONDEPU, T. H. BINDU and C. Z. BASHA,” Enhanced Computerized Classification System of Diseased Leaves,” 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 856-860 doi: https://doi.org/10.1109/ICECA49313.2020.9297413.
6. N. Jothiaruna and K. J. A. Sundar,” Survey on Diseased Leaf Using Segmentation,” International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 2019, pp. 225-230 doi: https://doi.org/10.1109/ISS1.2019.8908017.
7. A. Morbekar, A. Parihar and R. Jadhav,” Crop Disease Detection Using YOLO,” International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5 doi: https://doi.org/10.1109/INCET49848.2020.9153986.
8. K. Roy et al.,” Detection of Tomato Leaf Diseases for Agro-Based Industries Using Novel PCA DeepNet,” IEEE Access, vol. 11, pp. 14983-15001, 2023 doi: https://doi.org/10.1109/ACCESS.2023.3244499.
9. H. Amin, A. Darwish, A. E. Hassanien and M. Soliman,” End-to-End Deep Learning Model for Corn Leaf Disease Classification,” IEEE Access, vol. 10, pp. 31103-31115, 2022 doi: https://doi.org/10.1109/ACCESS.2022.3159678.
10. N. Bharanidharan, S. R. S. Chakravarthy, H. Rajaguru, V. V. Kumar, T. R. Mahesh and S. Guluwadi,” Multiclass Paddy Disease Detection Using Filter-Based Feature Transformation Technique,” IEEE Access, vol. 11, pp. 109477- 109487, 2023 doi: https://doi.org/10.1109/ACCESS.2023.3322587.
11. P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang,” Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks,” IEEE Access, vol. 7, pp. 59069-59080, 2019 doi: https://doi.org/10.1109/ACCESS.2019.2914929.
12. F. Liu and Z. Xiao,” Disease Spots Identification of Potato Leaves in Hyper-spectral Based on Locally Adaptive 1D-CNN,” IEEE International Conference on Artificial Intelligence and Computer

Applications (ICAICA), Dalian, China, 2020, pp. 355-358 doi: https://doi.org/10.1109/ICAICA50127.2020.9182577.

13. Nawaz, M., Nazir, T., Javed, A., et al. A robust deep learning approach for tomato plant leaf disease localization and classification. Sci. Rep. 12, 18568 (2022) https://doi.org/10.1038/s41598-022-21498-5.
14. P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi and S. G. Koolagudi,” Tomato Leaf Disease Detection Using Convolutional Neural Networks,” Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2018, pp. 1-5 doi: https://doi.org/10.1109/IC3.2018.8530532.
15. O. Kulkarni,” Crop Disease Detection Using Deep Learning,” Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4 doi: https://doi.org/10.1109/ICCUBEA.2018.8697390.
16. G. Shrestha, Deepsikha, M. Das and N. Dey,” Plant Disease Detection Using CNN,” IEEE Applied Signal Processing Conference (ASPCON), Kolkata, India, 2020, pp. 109-113 doi: https://doi.org/10.1109/ASPCON49795.2020.9276722.
17. Meeradevi, R. V, M. R. Mundada, S. P. Sawkar, R. S. Bellad and P. S. Keerthi,” Design and Development of Efficient Techniques for Leaf Disease Detection using Deep Convolutional Neural Networks,” IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Udupi, India, 2020, pp. 153-158 doi: https://doi.org/10.1109/DISCOVER50404.2020.9278067.
18. H. D. Gadade and D. K. Kirange,” Tomato Leaf Disease Diagnosis and Severity Measurement,” Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 318-323 doi: https://doi.org/10.1109/WorldS450073.2020.9210294.
19. R. G. de Luna, E. P. Dadios and A. A. Bandala,” Automated Image Capturing System for Deep Learning-based Tomato Plant Leaf Disease Detection and Recognition,” TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 1414-1419 doi: https://doi.org/10.1109/TENCON.2018.8650088.
20. H. Kibriya, R. Rafique, W. Ahmad and S. M. Adnan,” Tomato Leaf Disease Detection Using Convolution Neural Network,” International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 2021, pp. 346-351 doi: https://doi.org/10.1109/IBCAST51254.2021.9393311.
21. H. Xinru, C. Limin, X. Qing, L. Chongyuan, W. Yinchai and F. Peijun,” Plant disease detection based on improved YOLOv5,” 2nd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC), Mianyang, China, 2023, pp. 162-165 doi: 10.1109/RAIIC59453.2023.10280962.
22. A. Farhanah and W. F. Al Maki,” Hops Plants Disease Detection using Feature Selection based BPSO-SVM,” 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Jakarta, Indonesia, 2022, pp. 389-393 doi: https://doi.org/10.23919/EECSI56542.2022.9946620.

23. C. Gao and L. Liu,” C-YOLOX: Improved YOLOX Model for Tomato Diseased Leaves Recognition,” International Seminar on Computer Science and Engineering Technology (SCSET), New York, NY, USA, 2023, pp. 355-359 doi: https://doi.org/10.1109/SCSET58950.2023.00084.
24. A. Hidayatuloh, M. Nursalman and E. Nugraha,” Identification of Tomato Plant Diseases by Leaf Image Using Squeezenet Model,” International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 2018, pp. 199-204 doi: https://doi.org/10.1109/ICITSI.2018.8696087.
25. N. Schor, A. Bechar, T. Ignat, A. Dombrovsky, Y. Elad and S. Berman,” Robotic Disease Detection in Greenhouses: Combined Detection of Powdery Mildew and Tomato Spotted Wilt Virus,” IEEE Robotics and Automation Letters, vol. 1, no. 1, pp. 354-360, Jan. 2016 doi: https://doi.org/10.1109/LRA.2016.2518214.

Figures

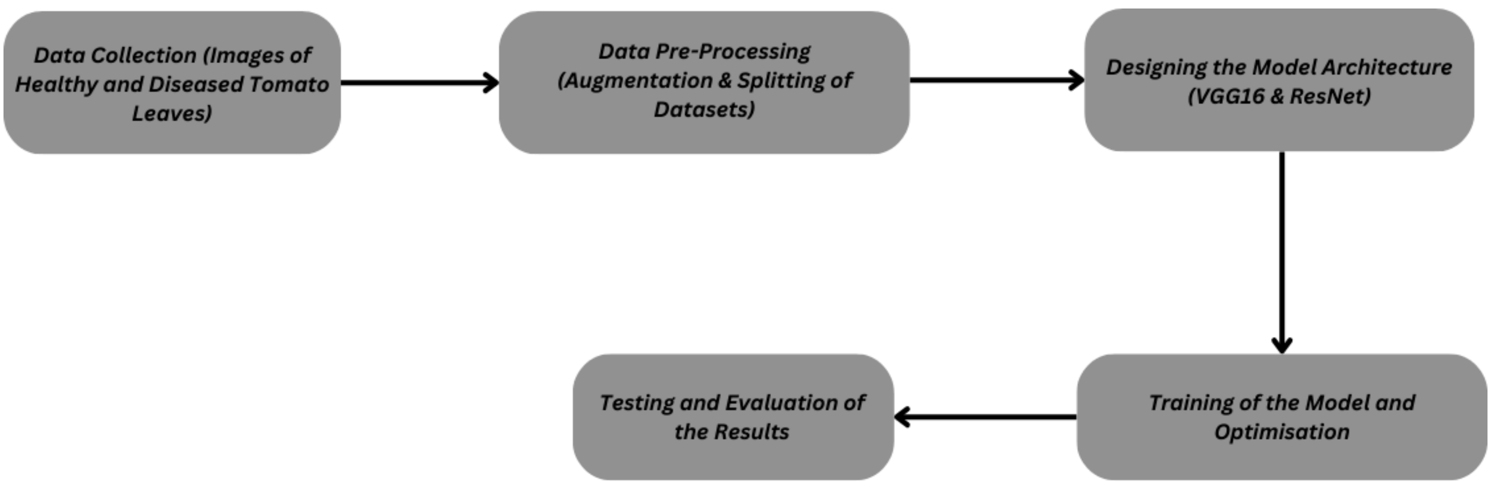

Figure 1

Various Blocks included in the Model Methodology.

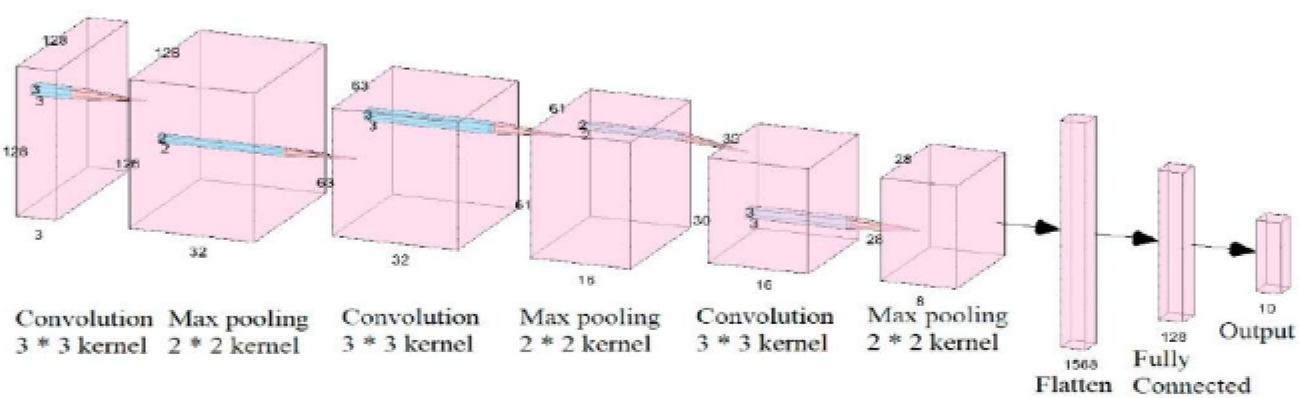

Figure 2

Basic CNN architecture visual representation

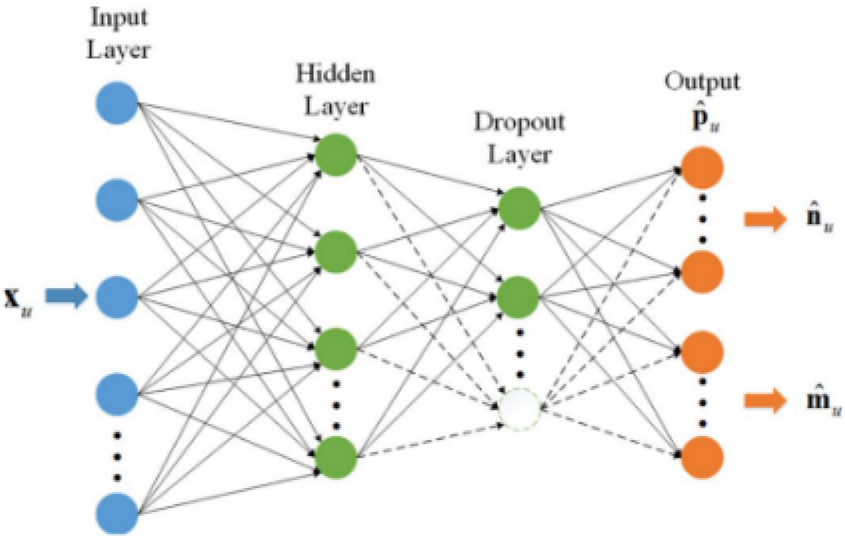

Figure 3

Architecture of Visual Geometry Group 16 (VGG16)

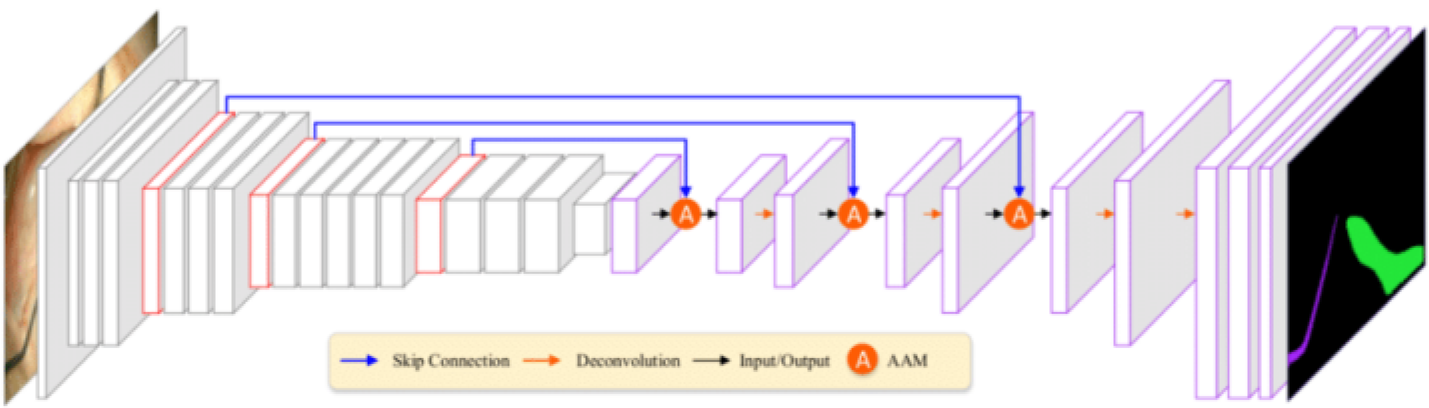

Figure 4

Architecture of Residual Network (ResNet)

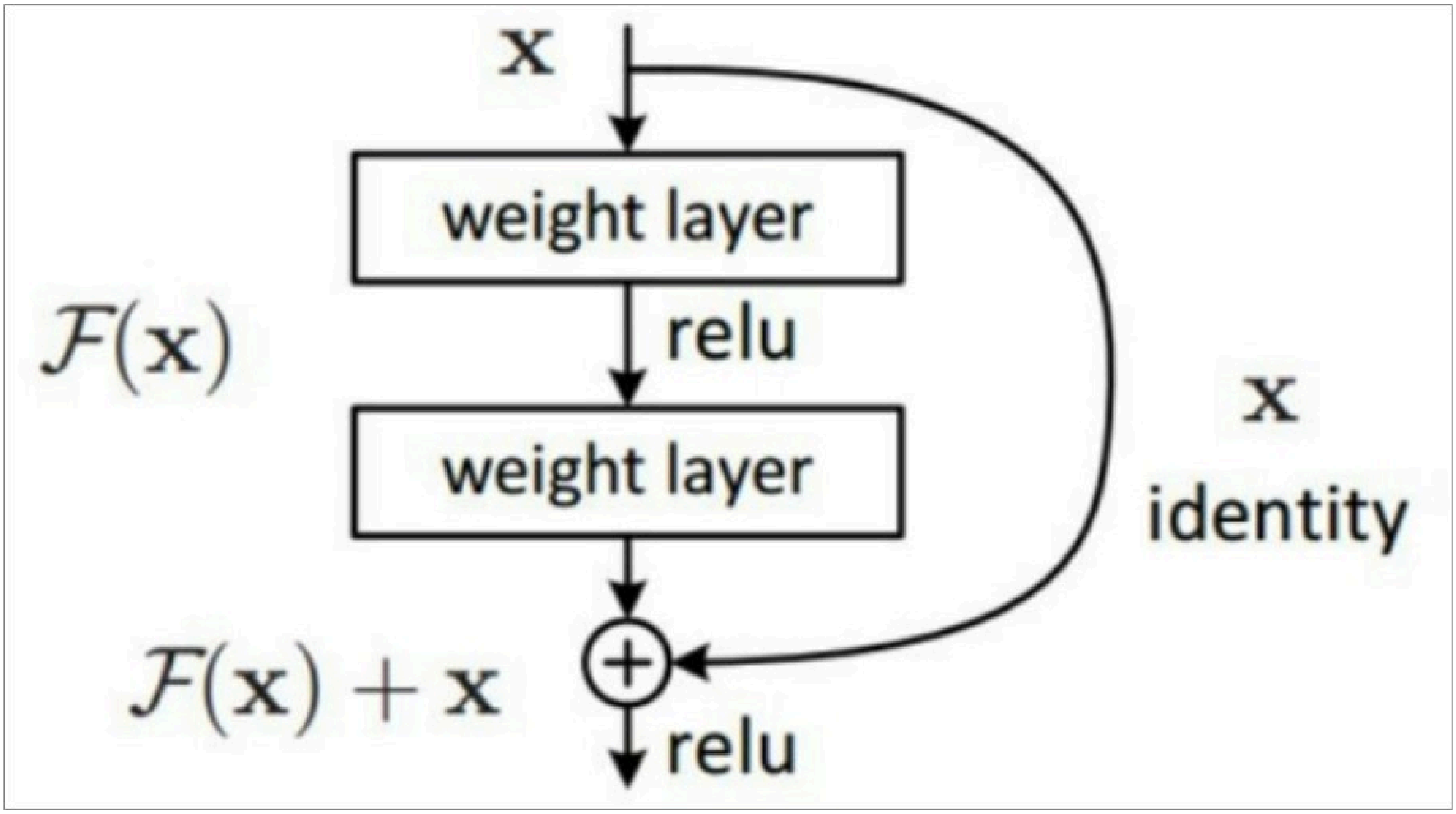

Figure 5

Building block of ResNet architecture

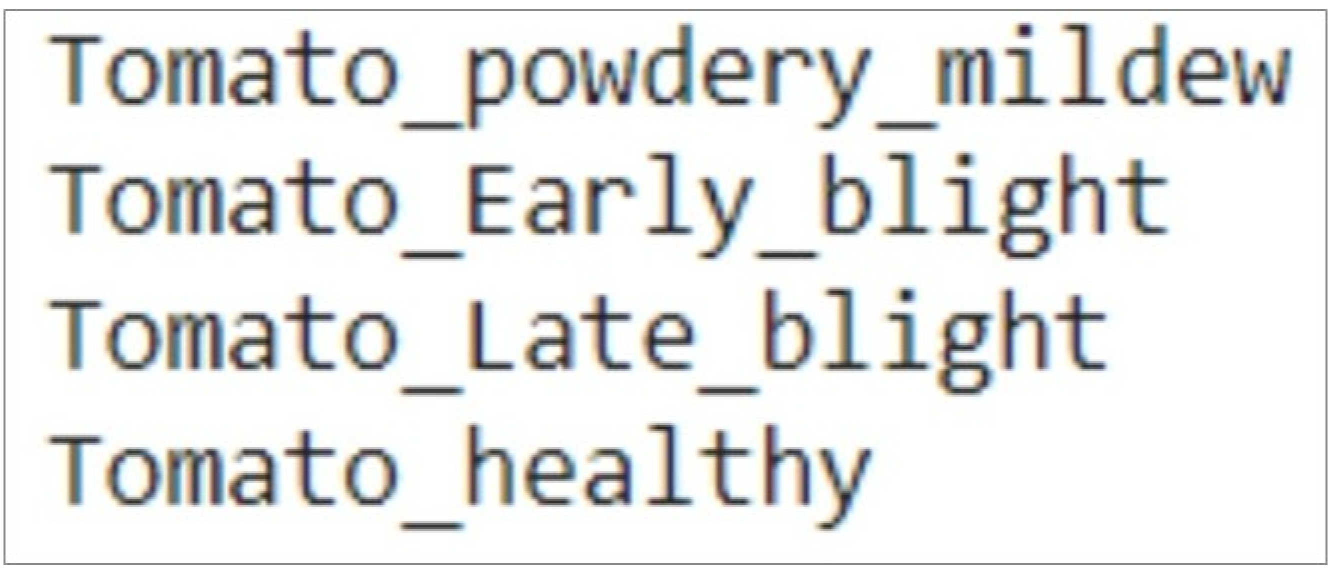

Figure 6

Dataset classes considered

Figure 7

Random Images of the Dataset classes considered

Figure 8

Learning rate of the model

66.67% [2/3 3:44:35<1:52:17]

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.692623	0.203530	0.930218	0.069782	1:53:02
1	0.324031	0.117249	0.956917	0.043083	1:51:33

61.67% [37/60 56:32<35:08 0.2246]

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.692623	0.203530	0.930218	0.069782	1:53:02
1	0.324031	0.117249	0.956917	0.043083	1:51:33
2	0.184742	0.089094	0.967233	0.032767	1:50:08

Figure 9

Accuracy and other parameters of the trained Model

CLASS	ACCURACY	# SAMPLES
bacterialspot	0.96	150
earlyblight	0.91	150
healthy	0.99	150
lateblight	0.95	150
leafmold	0.95	150

Figure 10

Accuracy of the various datasets considered

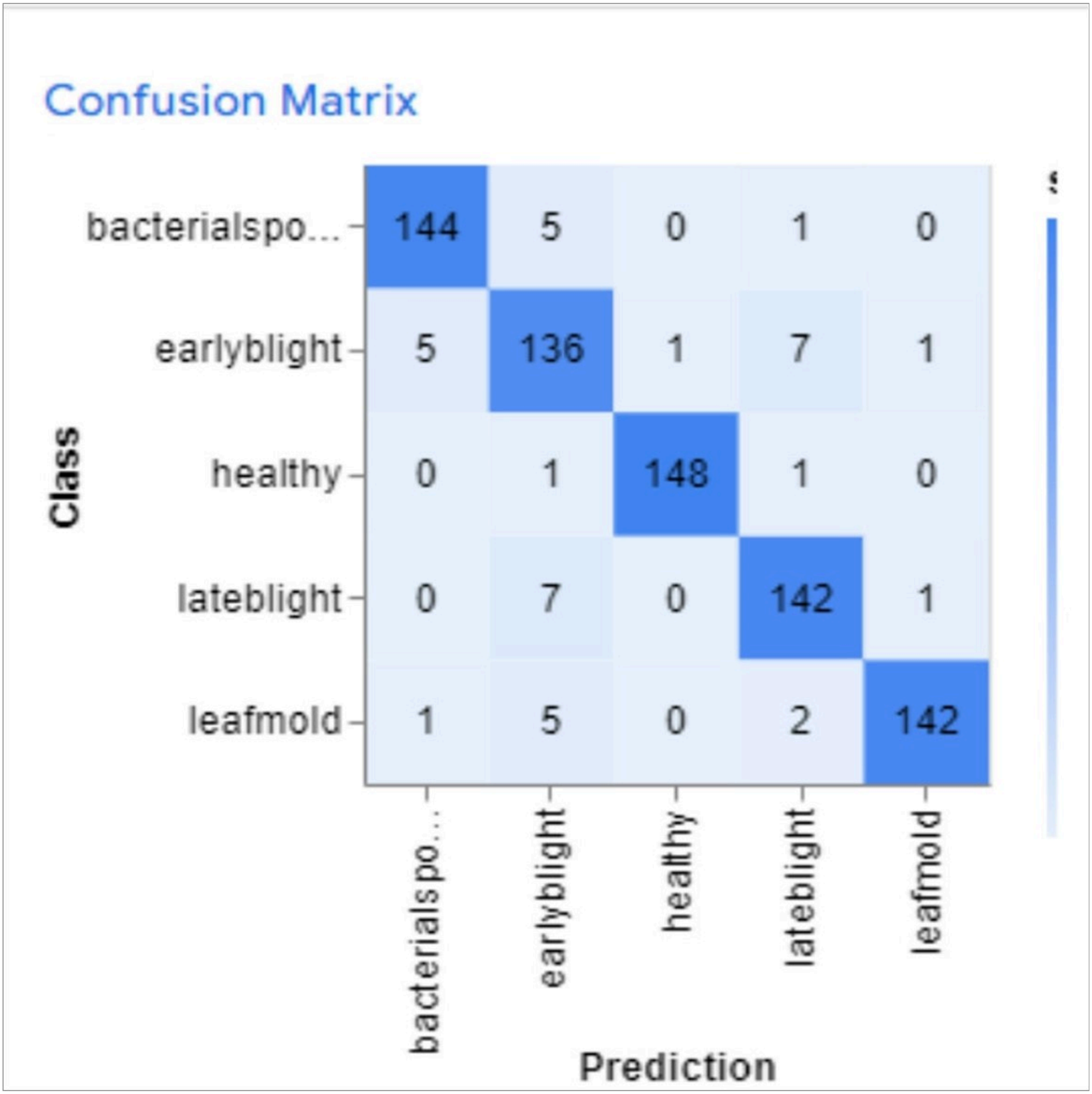

Figure 11

Accuracy and Error Matrix of the trained algorithms

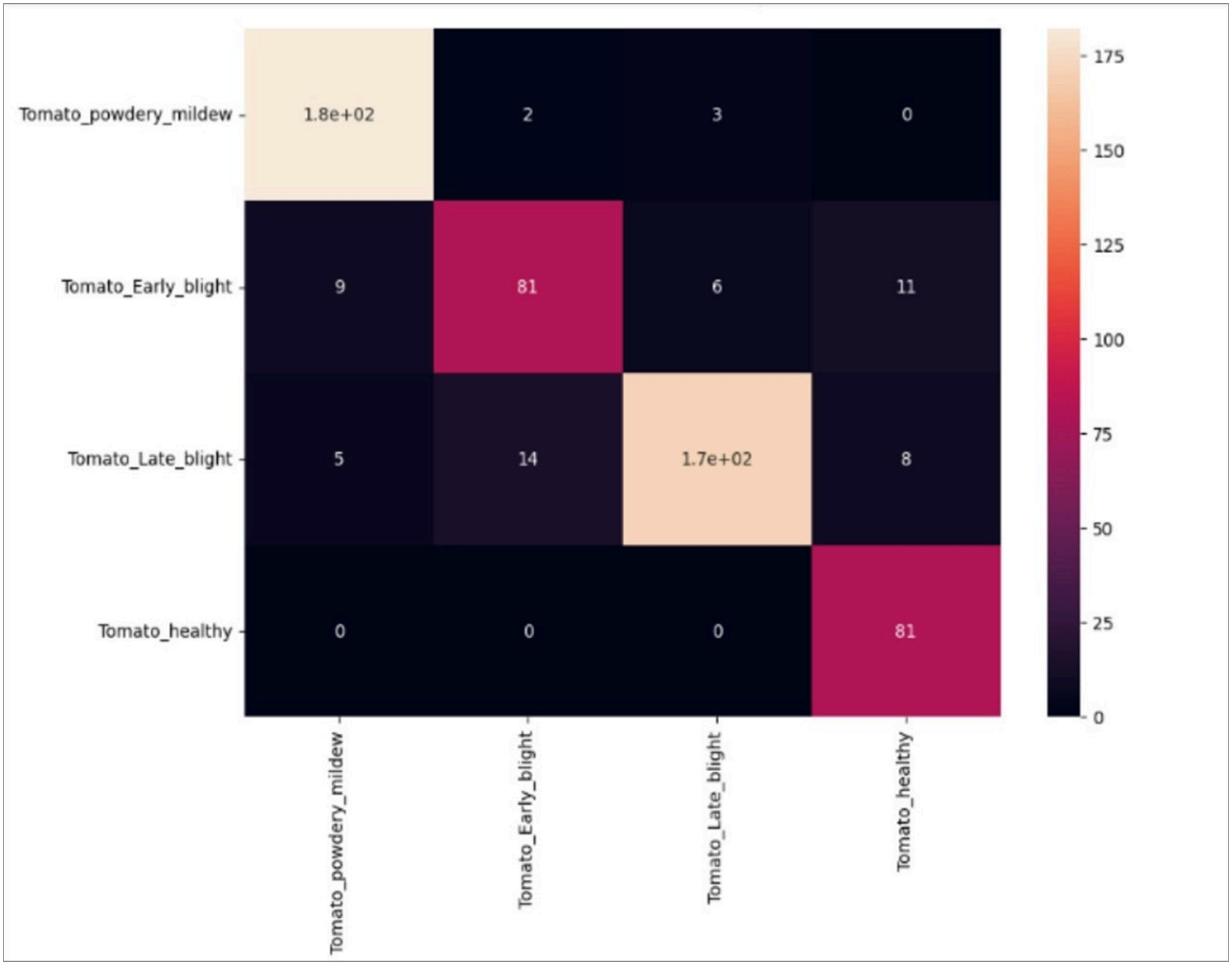

Figure 12

Heat Map of the proposed Model

Accuracy per epoch

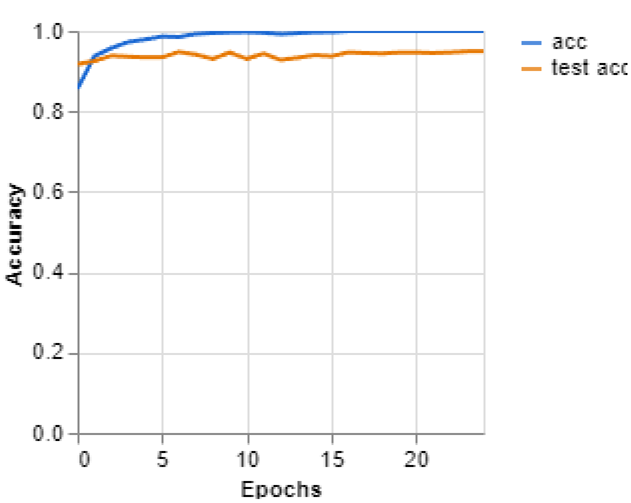

Figure 13

Accuracy of the trained system

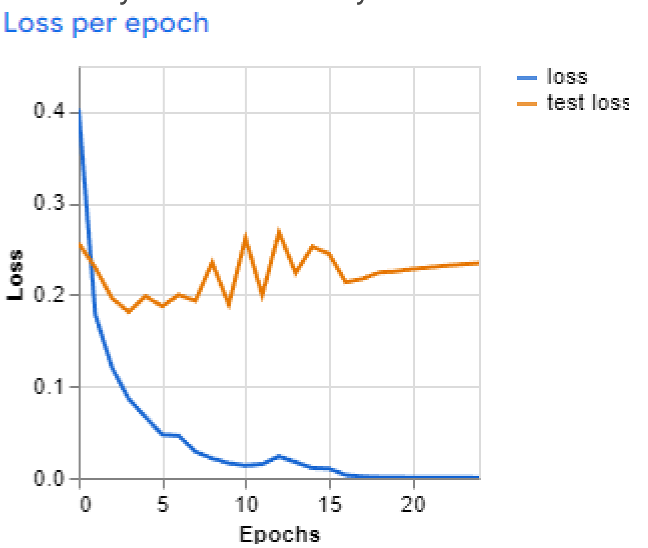

Figure 14

Loss of the trained system.

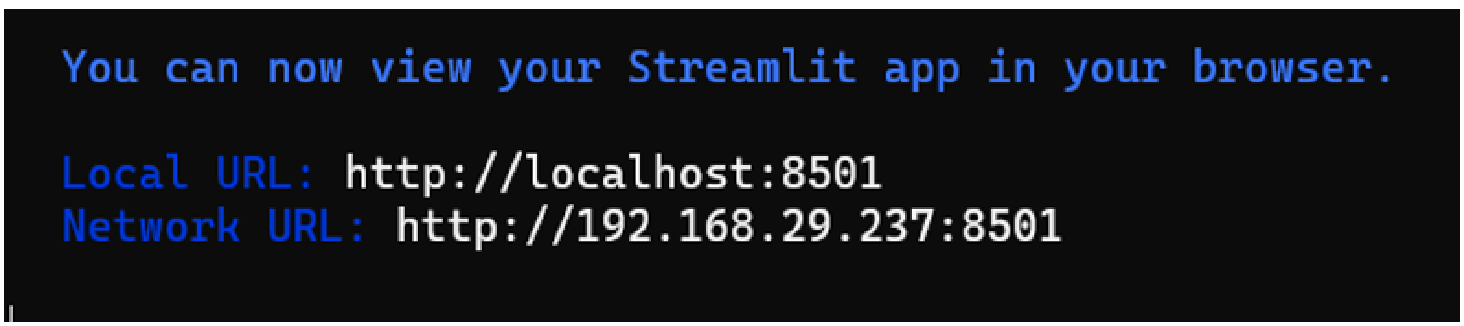

Figure 15

Unnumbered image in the Results and Discussions section.

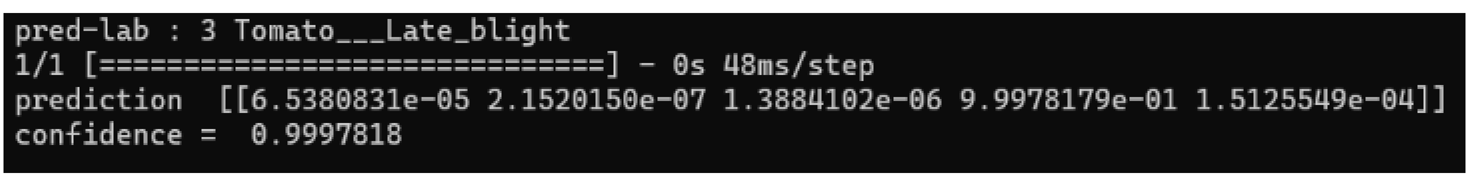

Figure 16

Unnumbered image in the Results and Discussions section.