LeADS (Leaf Anomaly Detection System):Deep Learning Pipeline for Leaf Stress, Disease & Severity Estimation

Mayur Pratim Das, Sushil Ghildiyal, Neeraj Goel and Mukesh Saini

[1] Department of Computer Science and Engineering
[2] Indian Institute of Technology Ropar, India
(2023csm1007, sushil.20csz0021, neeraj, mukesh)@iitrpr.ac.in

Abstract. Leaves are amongst the most sensitive parts of a plant. Any anomaly in the plant often affects the leaves first. Understanding the exact cause of leaf anomaly is a multi-faced challenge. We have come up with a novel end-to-end leaf anomaly detection system that differentiates between stressed and diseased leaf in the first stage using custom InceptionV3 model. Then it detects the exact type of disease and localizes the affected leaf area in the second stage using two YOLOv8 models. At last, it estimates the severity of the detected disease in the third stage using YOLO detections and IoU calculations. We have used Transfer Learning to use pre-trained weights. Our system is able to detect multiple anomalies on a single image. Classification task has 92% accuracy while Object Detection task and Severity estimation has about 80% accuracy.

Keywords: Leaf Anomaly, Transfer Learning, Classification, Object Detection

1 Introduction

1.1 Leaf Anomaly Detection

When it comes to detecting the anomalies in leaves one fundamental challenge that arises is determining the exact cause. If a plant leaf appears in bad shape or color, there might be multiple reasons for it. Presence of some pathogenic entity that causes diseases is often considered as the primary cause but similar effects are induced by stress events like absence or over abundance of water, nutrients, etc. Sometime is becomes very difficult to differentiate a diseased leaf from a stressed leaf as seen in [Fig. 1]

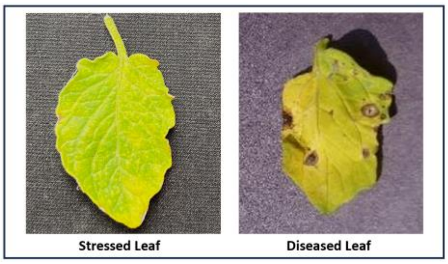


Fig. 1. Stressed vs Diseased Leaf: Minute differences

This can cause farmers to take wrong preventive measures and can further damage the plants. So, to correctly mitigate any leaf anomaly, differentiating between diseased leaves and stressed leaves becomes the preliminary task for proper anomaly handling.

1.2 Leaf Disease Detection

Once we are assured that the anomaly is caused by pathogenic disease, the next challenge that arises is determining the exact type of disease.

Common Leaf Diseases in Tomato

Tomato diseases can be caused by several parameters affecting the plants such as fungi, viruses, bacteria and environment. Leaf diseases directly affect the photosynthetic capacity of the plant and damages it. For our experiments we are studying four most common diseases encountered in Tomato Leaves: **Bacterial Spot, Early Blight, Late Blight and Septoria Leaf Spots.** [1]

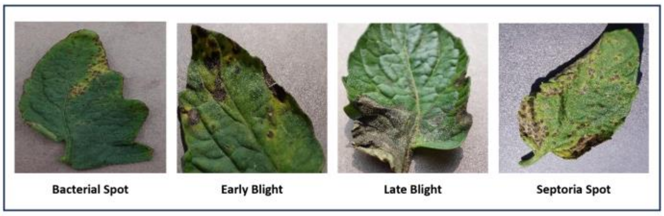


Fig. 2. Common Leaf Diseases in Tomato

It is very necessary to accurately detect the exact type of the diseases as each disease requires different mitigation measures.

1.3 Stage of Progression of Diseases

A very important factor that is often overlooked is that the damage a disease can due to a host organism is also heavily dependent on the stage or level of severity at which it is diagnosed. Most leaf diseases exhibit different stages of progression and the preventive measures also depend on the stage at which a disease is present. The following Fig. 3 depict how tomato leaf diseases can be clearly segregated on the basis of the stage or severity. Determining the stage of progression of disease can be a game changing step in leaf disease detection, which we try to do in our project.

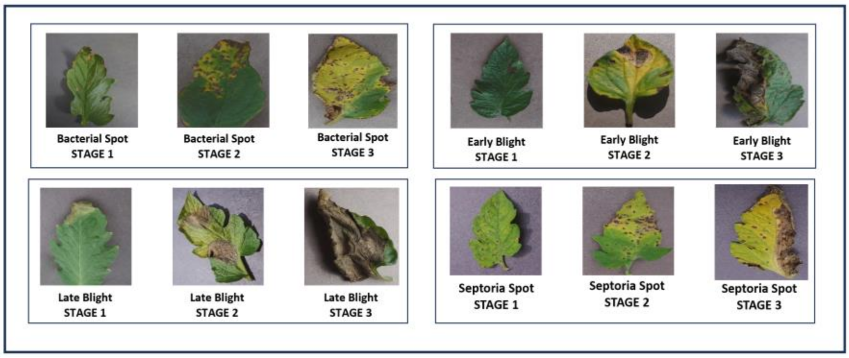


Fig. 3. Different Stages of Progression of Tomato Leaf Diseases

2 Related Work

Substantial research has been done in the field of anomaly detection of plant leaves in the recent years. Majority of the researches are involved in using different Machine Learning and Deep Learning techniques for leaf disease detection and classification [2] Speaking specifically about disease detection in Tomato leaves, we come across many research work that used various Computer Vision, Image Processing methodologies implemented through various Deep Learning techniques.[3] H.Al-Hiary et al, [4] proposed the use of k-means Clustering with Neural Network for both detection and classification of plant disease. It displayed precise accuracy between 83% and 94%. This approach had less computational demands but recognition rate was found to decline. Jia Shijie et al, [5] used Transfer Learning, a sub-branch of Deep Learning for disease and pest detection in tomato leaves. They used the pre-trained weights of VGG16 and SVM to extract features and detect 10 most common diseases and pest found in tomato leaves such as Bacterial Spot, Early Blight, Leaf Mould, Septoria Spot, etc. They used *Fine Tuning* to construct end-to-end classification model based on original VGG16 model. Model was trained using Keras/Tensorflow and achieved classification accuracy of 89%. Meanwhile, Mohammed Brahimi et al, have done a comprehensive analysis of different Transfer Learning algorithms like VGG19, ResNet-101 and MobileNet-v2 on PlantVillage[6] and CCMT [7] datasets to classify leaves based on various common tomato leaf diseases. The best performance was achieved by VGG19, where the best accuracy, precision, recall, and F1-score on the test set of PlantVillage and CCMT datasets were 99.48%, 99.27%, 99.28%, 99.27%, and 92.76%, 92.74%, 95.09%, 90.86%, respectively. All these research work focuses on classification of the leaf images based on the disease present on the leaf. Although they have achieved fairly good results on classification, they do not focus on localization or finding out the pinpoint location of the leaf anomaly. For stress-related anomaly, it might not be necessary to precisely localize the anomaly but in case of presence of pathogenic disease related anomaly it may be beneficial.

Stanley G. E. Brucal et al, [8] have decided to go with YoloV8 Object Detection model to detect type of disease and localize it. YoloV8 model has proven its high potential to classify tomato leaf disease when it resulted in a mean Average Precision (mAP) of 98.9%, at an average precision rate of 97.5% and recall rate of 91.9%. They have not considered differentiating between Stressed and Diseased leaves. G. Priyadharshini et al, [9] have tried to detect and localize tomato leaf diseases by using CNN, R-CNN, Fast R-CNN and Faster R-CNN and have done comparative study between them. Faster R-CNN provided the best results with accuracy around 98%. The aforementioned two literatures have achieved disease detection and localization with a fairly good accuracy. However, all of the above discussed researches have not addressed an important aspect of disease detection which is the stage or severity of the disease.
Sanjay B. Patil et al, [10] used simple Digital Image Processing techniques to segment the diseased areas on sugarcane leaves and assign a severity of the manifested disease on a 5-level severity index based on lesion area. Caihua Yao et al, [11] have used a combination of Object detection using YOLO and Segmentation using U-Net based

architecture to estimate the stage or severity of the disease in plum leaves. All these papers do not consider the need of differentiating between stressed and diseased leaves before diving into disease detection and severity estimation.

Some common research gap has been observed among all the literature discussed above. For a complete end-to-end Anomaly Detection setup for a leaf, we must solve a 3-faced problem. Firstly, to determine if the anomaly is caused due to stress or due to pathogenic disease. Secondly, if we are assured that a pathogenic disease is involved, need to accurately detect and localize the type of disease manifested. Thirdly, estimate the stage or severity of the disease. To the best of our knowledge, there doesn't exist any literature that addresses all the three challenges at once. This paper aims to address this research gap and provide a complete solution of detecting leaf anomalies in real world environments.

3 Problem Definition

The challenge of anomaly detection in Tomato leaves requires us to tackle a 3-front problem. First challenge is to determine if the visible anomaly is due to disease or due to stress. Second challenge is to determine the exact type of disease that has manifested onto the leaf. Third challenge is to determine how much the disease has progressed, i.e. determining the stage of progression of the disease. Addressing this 3-faced challenge required us to develop a multi-stage robust Deep Learning pipeline that can successfully tackle all these challenges independently and accurately and provide a detailed overview of the anomaly in the tomato leaves.

4 Dataset

We have searched various datasets across various platforms and created three custom datasets from a subsection of the datasets Dataset for Crop Pest and Disease Detection [12] and A Dataset for Visual Plant Disease Detection [13]. The following section describes each of the three datasets in details.

4.1 MPD-Iv3 Classifier Dataset

This dataset contains tomato leaf images segregated into three classes:

1. *Tomato_Healthy :* This contains images of heathy, stress-free and disease-free tomato leaves and plants.

2. Tomato_Stressed : This contains images of stressed tomato leaves like yellowed leaves and curled leaves.

3. Tomato_Diseased : This contains a rich variety of tomato leaves with wide variety of diseases.

Distribution of the dataset is depicted in the Fig. 4 below.

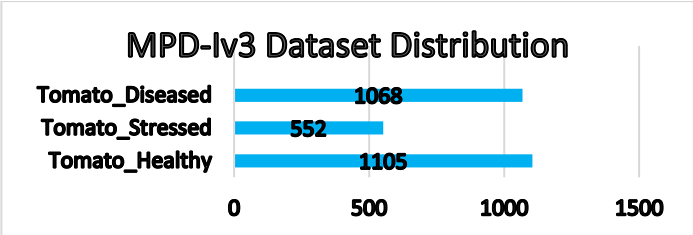


Fig. 4. Dataset Distribution of MPD-Iv3 Dataset

This dataset will be used for our STAGE I of leaf analysis which involves classification through custom **Inception v3 Model**.

4.2 MPD-Yv8 Leaf Detector Dataset

This is our second custom dataset. It contains around 100 annotated images of leaf boundaries. It contains good mix of different diseased leaves with the "Leaf Area" demarcated explicitly. So, it contains a single class ***Leaf Area*** for detecting tomato leaves. This dataset is used in STAGE II of our pipeline where we use Yolov8 model **Leaf Detector** to detect individual leaves.

4.3 MPD-Yv8 Disease Detector Dataset

This is out third dataset that contains annotated images of diseased areas of leaves. We have annotated for 4 diseases and each disease have been segregated into 3 types A, B and C based on difference in looks and features. This dataset has 12 annotated classes:

Class Name	***Description***
Bacterial Spot_A	Small, dark, water-soaked lesions on the leaves.
Bacterial Spot_B	Lesions enlarge and develop into irregularly shaped spots with a yellow halo.
Bacterial Spot_C	Lesions coalesce, causing severe damage to the leaves, and leading to leaf defoliation
Early Blight_A	Small, dark brown spots with concentric rings on the lower leaves of the tomato plant.
Early Blight_B	Lesions expand and coalesce, leading to the yellowing and wilting of the affected leaves.
Early Blight_C	Extensive defoliation occurs, severely impacting the yield and health of the tomato plant.
Late Blight_A	Dark, water-soaked lesions on the leaves, fuzzy white mould growth on the underside.
Late Blight_B	Lesions rapidly expand, turning dark brown to black, and causing severe damage to the foliage.
Late Blight_C	Lesions continue to spread, affecting not only the leaves but also the stems and fruits, leading to plant death in severe cases.
Septoria Spot_A	Small, circular lesions with dark brown margins and grey centres on the lower leaves.
Septoria Spot_B	Lesions enlarge and coalesce, leading to extensive browning and defoliation of the lower leaves.
Septoria Spot_C	Lesions spread to the upper leaves and stems, severely affecting the overall health and productivity of the tomato plant.

Table 1. Description of each class of MPD-Yv8 Disease Detector Dataset.

This dataset will be used in STAGE II of our pipeline which involves detecting the diseased areas in leaves by **Disease Detector** Yolo model and also in STAGE III for determining the stage of progression of the disease.

It is to be noted that the reason each disease class has been further segregated into three types A, B and C is that this segregation will act as Ground Truth for Disease Stage or severity estimation that occurs at out STAGE III of LeADS. Careful annotation is done by annotating the diseased area in each leaf. Also, each leaf has been segregated into three types pertaining to three severity stages. Each class has close to 100 annotated images so the total dataset consists more than 1200 annotated images of four tomato leaf disease segregated on basis of severity as well.

5 Methodology

In this paper we propose **Leaf Anomaly Detection System (LeADS)**, a Multi-Stage Deep Learning pipeline capable of providing end-to-end analysis of Tomato Leaves. This pipeline has been designed keeping in mind keeping three challenges or tasks that we wish to tackle in each stage. Next sections describe each of these stages in details.

5.1 STAGE I: Classification between Healthy, Stressed and Diseased.

As described before, the preliminary challenge in correctly diagnosing any leaf anomaly is to differentiate between ***Diseased*** and ***Stressed*** leaves. We need a robust Deep Learning model capable of performing this classification task very precisely. After experimenting with different deep learning models like ResNet-50 and VGG16 (results discussed later), we decided to go with **InceptionV3 model**. Its architecture consists of multiple Inception modules that apply different types of convolutions (e.g., 1x1, 3x3, 5x5) and pooling operations in parallel. This allows the network to learn both fine and coarse features, which is essential for accurately classifying leaves that might have subtle differences in appearance due to disease or stress. It is also very useful for Transfer learning, i.e. use it as pre-trained model on large datasets like ImageNet [14] which can be fine-tuned to the specific task of classifying tomato leaves. This is particularly useful as our dataset is not extremely large.
For our Stage I we setup our custom Inceptionv3 architecture in the following way:

1. **Input Image Preprocessing:** Set the image input size to 224x224 pixels, a standard size that balances computational efficiency with model accuracy.
2. **Model Architecture Customization:** For Transfer learning we do the following-

    - *Base Model:* Loaded the InceptionV3 model with pre-trained weights from ImageNet, excluding the top layers. This allows the model to be fine-tuned for the specific task of classifying tomato leaves.
    - *Layer Freezing:* All layers of the InceptionV3 model were frozen to retain the pre-trained weights, avoiding unnecessary modifications that could destabilize the learning process.
    - *Adding Custom Layers:* Flattened the output of the InceptionV3 model to convert the multi-dimensional feature maps into a one-dimensional vector. Dense (fully convoluted) Layer is also added with Softmax activation the end for output probabilities for each output class.

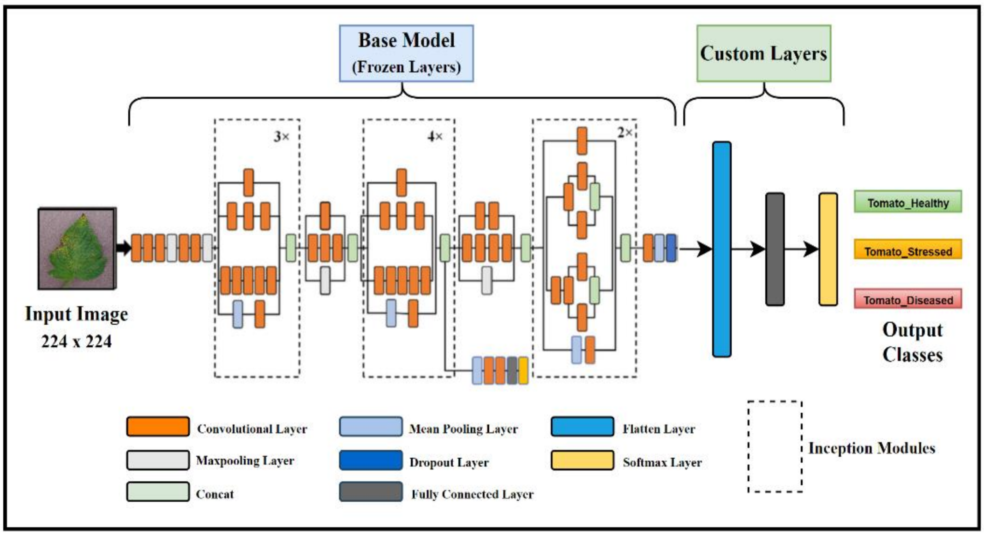


Fig. 5. STAGE I (Classification) Architecture using custom Inceptionv3 Model.

For our pipeline, we first prepare the MPD-Iv3 dataset as discussed above and use it to train out Inception v3 model. We get output among the following three classes:

Table 2. Stage I (Classification) outputs and corresponding actions.

Output Class	Inference	Action Taken
Tomato_Healthy	Input image is disease free and stress free.	Pipeline halts.
Tomato_Stressed	Input image is indicative of a stressed plant leaf, and not diseased. Cause of stress is of future work for our pipeline.	Pipeline halts.
Tomato_Diseased	Input leaf image is indeed, suffering from some pathogenic disease.	Stage II of pipe-line activates

5.2 STAGE II: Object Detection for Disease Type Identification

Our next task is to precisely determine the type of disease on the leaf. Input of this stage is a diseased leaf image and expected output is one of the classes of disease type described in MPD-Yv8 dataset and also bounding box coordinates of the diseased/affected area. There may exist more than one disease type in multiple leaves in a single input image and on top of that, there may be leaves with same disease but different stage of progression. Hence, we decided to go with YOLO capable of simultaneous multi-class object detection model. This model was developed by Ultralytics.[15] and currently one of the most popular models for object detection.

We train two YOLO Models:

Leaf Detector: An input image may contain multiple leaves and each leaf may contain different disease/anomaly. Hence, we need to isolate the individual leaves first. Trained on MPD-Yv8 Leaf Detector Dataset, the Leaf Detector model detects individual leaves. This model gives bounding box coordinates of individual leaves in the input image. We call this '*Leaf Area*' We then crop these portions of the input image and pass it on to the Disease Detector Model.

Disease Detector: Trained in MPD-Yv8 Disease Detector Dataset that detects the diseased parts of the leaf. This model takes individual cropped portions [*Leaf Area(s)*] as input and gives bounding box coordinates of detection(s) in 'xyxy' format along with labels stating the Disease type (one of the 12 classes of MPD-Yv8 Disease Detector Dataset) as output. There may be multiple diseased areas and even different diseases on same Leaf Area. Our Yolov8 model is capable of detecting all these simultaneously.

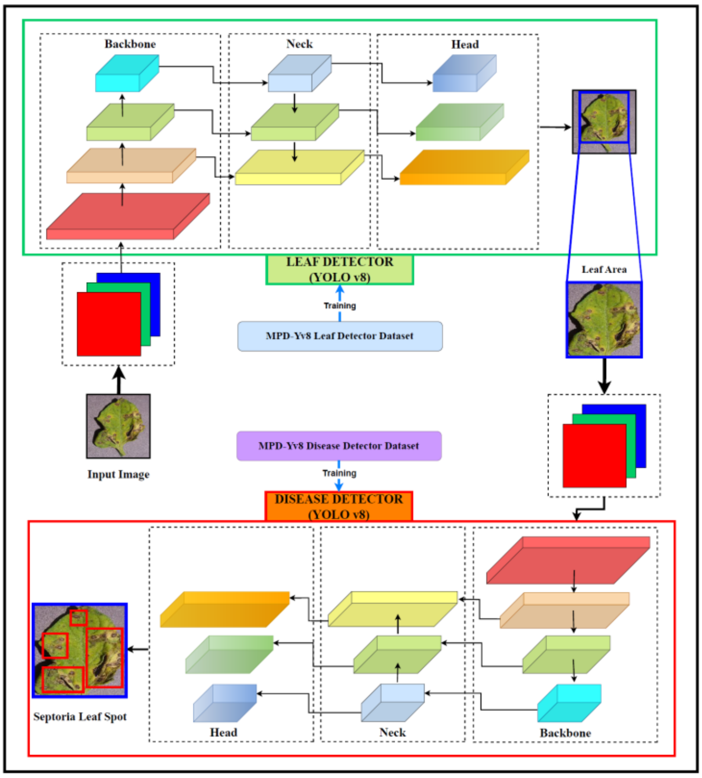


Fig. 6. STAGE II: Disease Detection using two Yolov8 models.

5.3 STAGE III: Disease Stage/Severity Calculation

This is the final task of **LeADS** pipeline where we calculate the Severity stage of the detected disease. We take into consideration the total surface area of the covered by the disease to determine the severity of the anomaly. The process of Severity stage calculation is described below:

Step 1: Intersection over Union (IoU) Calculation: For each leaf $\boldsymbol{L_i}$ detected in the image, we have a set of diseased areas $\boldsymbol{D_j}$ detected by the Disease Detector model. The IoU between the leaf area $\boldsymbol{A}(\boldsymbol{L_i})$ and each diseased area $\boldsymbol{A}(\boldsymbol{D_j})$ is calculated as:

$$\boxed{IoU(L_i, D_j) = \frac{A(L_i) \cap A(D_j)}{A(L_i) \cup A(D_j)}}$$

Where:

- $A(L_i)$ is the area of the *i*-th leaf.
- $A(D_j)$ is the area of the *j*-th Diseased region.
- $A(L_i) \cap A(D_j)$ is the intersection area between leaf and the diseased region.
- $A(L_i) \cup A(D_j)$ is the union area of the leaf and the diseased region.
- **Step 2: Summing Diseased Areas:** For each detected leaf L_i, sum the IoUs of all overlapping diseased regions D_j with that leaf:

$$\boxed{Total\ IoU(L_i) = \sum_j IoU(L_i, D_j) \ \ \textbf{for all } j \textbf{ where } IoU(L_i, D_j) > 0}$$

This gives the Total IoU, representing the overlap between the leaf and the diseased areas. Let us call it **Infection Score**, I_S

Step 3: Disease Stage Determination: Once the total IoU is computed, the disease stage or ***Severity Stage***, S_s for the leaf L_i is determined based on predefined thresholds T_1 and T_2:

$$\boxed{S_s = \begin{cases} Stage\ 1\ (Mild) & if\ \ I_S \leq T_1 \\ Stage\ 2\ (Moderate) & if\ \ T_1 < I_S \leq T_2 \\ Stage\ 3\ (Severe) & if\ \ I_S > T_2 \end{cases}}$$

Where:

- T_1 and T_2:is the threshold for transitioning from Stage 1 to Stage 2.
- T_2 is the threshold for transitioning from Stage 2 to Stage 3.

The Threshold values T_1 and T_2 are changeable depending on specific requirements but they are set to $T_1 = 0.15$ and $T_2 = 0.4$ for all out experiments. Leaf Area $A(L_i)$ and Diseased Areas $A(D_j)$ are calculated using the Bounding Box coordinates given by *Leaf Detector* and *Disease Detector* models respectively [Fig. 6].

5.4 Final Algorithm for Leaf Anomaly Detection System (LeADS)

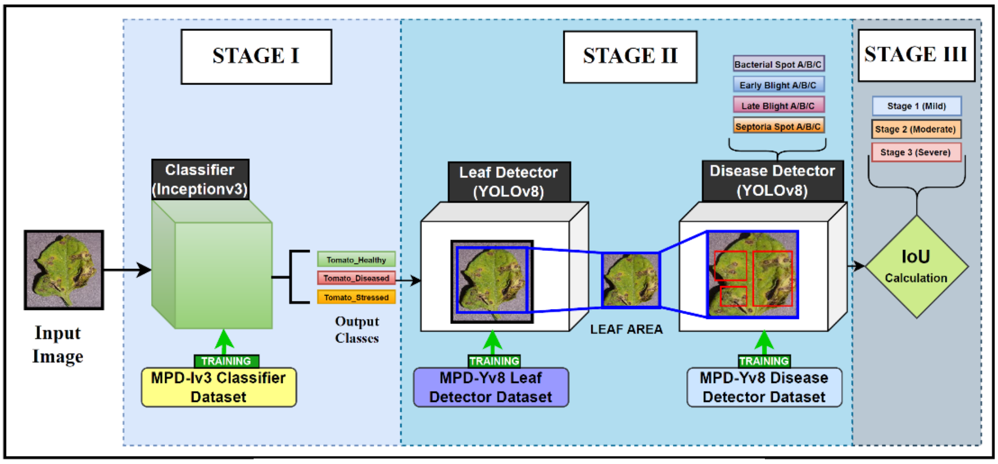


Fig. 7. LeADS Multi-stage Anomaly Detection Pipeline

Algorithm 1 Leaf Anomaly Detection System **(LeADS)**
Require: • Coloured image of Tomato Leaves • InceptionV3_model: Pre-trained InceptionV3 model for Stage 1 classification. • YOLOv8_LeafDetector: Pre-trained YOLOv8 model for detecting leaves in Stage 2. • YOLOv8_DiseaseDetector: Pre-trained YOLOv8 model for detecting diseased areas in Stage 2. • T_1, T_2: Predefined IoU thresholds for disease stage classification.
Ensure: Detection of Stress or Disease Type with Localization & Severity estimation

1. **Preprocessing:** Resize input image into 224 x224 and normalize pixel values.
2. **STAGE I:** Differentiate between Healthy, Stressed or Diseased leaf using custom InceptionV3 model to remove ambiguity between disease and stress.
3. ***if* (Predicted Class == 'Tomato_Healthy' || 'Tomato_Stressed')**
    Return
4. ***else if* (Predicted Class == 'Tomato_Diseased') *do:***
    • **STAGE II:** Detect disease type and localize the affected area using two YOLOv8 Models, **Leaf Detector and Disease Detector**
        ○ **Leaf Detector:** Get Bounding Box coordinates of *Leaf Area(s)* L_i, crop the leaf area and pass as input to Disease Detector.
        ***Return* (Leaf Area(s) L_i)**
        ○ **Disease Detector: *for all*** Leaf Area(s) L_i ***do:***
            ▪ Detect diseased areas, identify disease class and return bounding box coordinates of Diseased Areas with class names of diseases.
        ***Return* (Disease Areas D_j, Disease Classes C_j)**
    • **STAGE III:** Calculate Disease Stage/Severity using L_i, D_j, T_1 & T_2.
    ***Return* (Disease Severity Stage: *Ss*)**
5. ***End***

6 Experimental Results

6.1 Experimental Setup

The proposed system has been implemented using Python 3.8 in Google Collab environment. A consistent hardware setup has been maintained for all experiments. The experimental hardware setup of Collab comprises of Intel Xeon CPU@ 2.20 Ghz with 2 vCPUs (virtual CPUs) and 13GB of RAM along with Nvidia T4 Tesla T4 GPU with 16 GB of VRAM. The datasets were split into training, validation and test set in typical 70:20:10 ratio. Since our pipeline consists of three Stages, we are going to discuss the experimental results of each Stage in details in the following sections.

6.2 STAGE I Results: Classification with Inception v3 on MPD-Iv3

For our Stage I we have used various Preprocessing and Data augmentation techniques. We have resized our images to 224 x 224 pixels. We have used data augmentation to normalize pixel values by rescaling the by 1. /255. We have also done shear, zooming and horizontal flip to induce variation in our data. For training, we have used Adam optimizer with learning rate of 0.001. Categorical Cross Entropy was chosen as Loss Function. We train our custom Inception model over 10 epochs with batch size of 32. Accuracy observed after training our Inceptionv3 Classification Model max accuracy achieved was 92% during multiple runs.

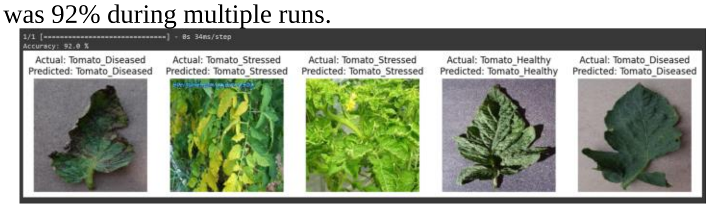


Fig. 8. Predictions with InceptionV3 (Classification)

We plot the Training and Validation Accuracy and Loss against number of epochs and see that the Validation Accuracy increases with increasing epochs. Also, the Validation Loss seems to decrease gradually with epochs.

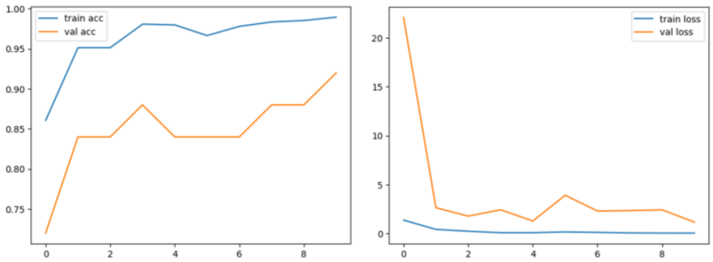


Fig. 9. Plots of Accuracy and Loss over 10 Epochs (custom InceptionV3 Model)

Next, we decided to use few other popular models on our same dataset to make a comparative analysis and to justify the use of Inception v3 model for our classification task We used ResNet-50 and VGG16 with the same MPD-Iv3 Dataset and the comparative results are shown in the table below:

Model	**Total Trainable Parameters (Millions)**	**Maximum Accuracy Achieved**
ResNet-50	30.1059	65.38 %
VGG-16	7.5267	80.77 %
Custom InceptionV3	15.3603	**92 %**

Table 3. Comparative study of different Deep Learning Models for STAGE I classification for differentiating between Healthy, Diseased and Stressed leaves.

The above results shows that Inceptionv3 serves as the best model for our classification task using transfer learning. The achieved results show that our STAGE I can confidently differentiate between Diseased and Stressed leaves thereby removing ambiguity.

6.3 STAGE II Results: Detection of Disease type and Localization of diseased area using two YoloV8 models.

The Yolov8 models uses transfer learning and we have done multiple runs to ensure proper convergence. As preprocessing we have resized all images to 256 x 256 and convert images to 'RGB' channels before providing as input to Leaf Detector or Disease Detector. STAGE II involves performing Object Detection in two phases.
First phase uses the MPD-Yv8 Leaf Detector dataset to train our Leaf Detector YoloV8 Model to detect 'Leaf Area' We have run our Yolo v8 model over 50 epochs on the dataset. The datasets were annotated with labelImg.

Fig. 10. Leaf Area Detections by Leaf Detector Yolo Model

The Leaf Detector is easily able to detect the 'Leaf Area' in the images with great accuracy as seen in Fig. 10. Next phase of STAGE II involves cropping all the 'Leaf Area's from input image and passing it into the Disease Detector Model.

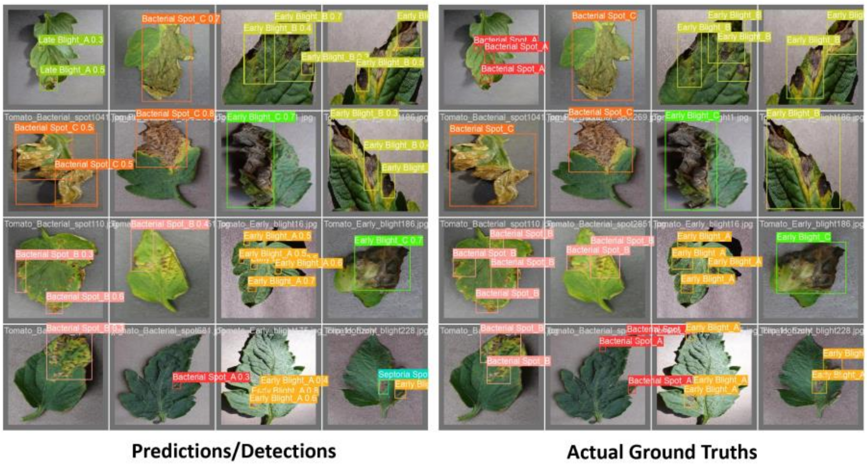


Fig. 11. Detections by Disease Detector Model

The above figure shows that the Disease Detector is able to correctly detect the disease labels in most cases. However, not all diseased areas are detected in all the leaves.

Analyzing the Confusion Matrix [Fig. 12] of the Disease Detector tells us that although most of the Disease labels are detected correctly but some labels like Bacterial Spot_A and Bacterial Spot_B are often not detected. It may be because labels marked A and B indicate smaller areas or stages of less severity, so all small areas may go undetected.

This issue can be observed in Fig. 11 as well in case of labels like Bacterial Spot_A where not all the small infected patches are detected. It may be due to human error in annotation as well.

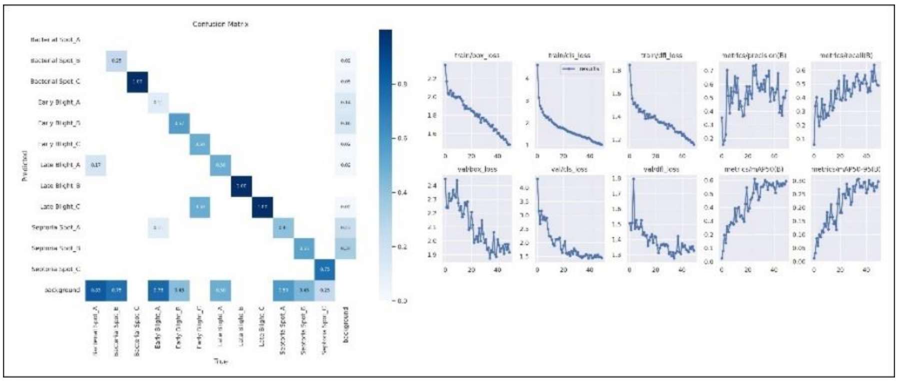


Fig. 12. Confusion Matrix of Disease Detector

However, it is worth noting that the Type of disease is always precisely detected. It thus serves its purpose of detecting the type of disease that is manifested on the leaf. We further evaluate its performance through various plots as shown in Fig. 12. The mAP50 and mAP95 plots indicate the similar issue of localization in few disease labels.

6.4 STAGE III Results: Disease Stage or Severity Estimation

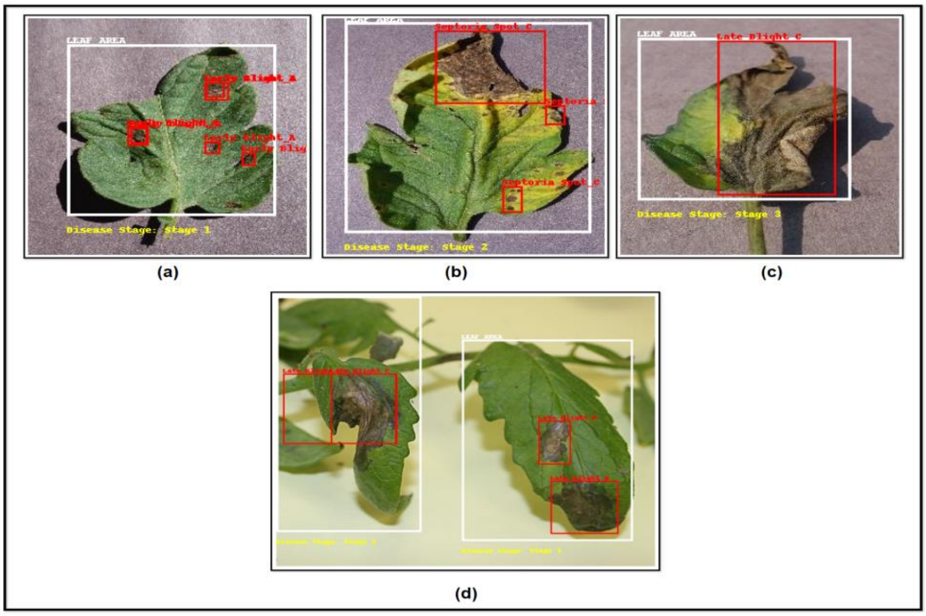
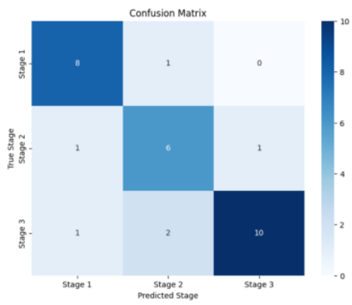


Fig. 13. Final outputs of **LeADS**.

The final stage of our pipeline involves estimating the Stage or Severity of the disease on the leaves. The process of determining the ***Infection Score***, $\boldsymbol{I_S}$ is already discussed. To evaluate the accuracy of these scores, we use the following method. We map the last letter of the label of the dominant detection in a leaf to a corresponding Stage and consider it as Ground Truth (True Stage). **A denotes Stage 1**; **B denotes Stage 2** and **C denotes Stage 3**. For e.g. **Septoria Spot_A** denotes Stage 1 detection of Septoria; **Early Blight_B** denotes Stage 2 detection of Early Blight and **Late Blight_C** denotes Stage 3 detection of Late Blight. Now we use around 30 images from validation set and run our pipeline on each of them and obtained the calculated severity stage (Predicted Stage).



Disease Severity Stage Classification Report			
Stage	**Precision**	**Recall**	**F1-Score**
Stage 1	0.80	0.89	0.84
Stage 2	0.67	0.75	0.71
Stage 3	0.91	0.80	0.80

Fig. 14. Confusion Matrix and Precision, Recall, F1 Scores of the Stages of Severity

After that we evaluate Confusion Matrix[Fig 14] for them and also calculate the overall accuracy and various other matrices. The Accuracy of **80%** was calculated in the classifications.

On analyzing the Confusion Matrix above we see that Stage 1 and Stage 3 severity cases are easily and accurately calculated. But Stage 2 is often misclassified as Stage 1 and Stage 3. Its primary reason is that the threshold values for Stage 2 lies between 0.15 to 0.4 which is a small window. So, the Stage 2 cases may frequently silp into Stage 3. Overall, this approach seems to do a fairly good job in determining the severity of a manifested disease on a leaf.

7 Discussion and Conclusions

Use of InceptionV3 for Classification provided satisfactory results. On the object detection front, YOLO models were quite robust in predicting the disease class but the Disease Detector struggled a bit when it came to pin-point localization of the diseased area, resulting in average mAP scores. Possible reason maybe human error in annotation and humble size of dataset. However, another possible reason can be that trying to bound irregular shaped areas like leaf shape and disease lesions with a rectangular bounding box, which resulted in skewed IoU calculations in some cases. Using segmentation to demarcate the diseased area and then try to predict the severity maybe a better option. This remains a promising direction of work in the future. Keeping this aside, LeADS showcased a novel approach on how to leverage Transfer Learning to provide an end-to-end solution for anomaly detection by solving a 3-faced challenge of removing ambiguity between diseased and stressed leaves, disease detection and localization as well as disease severity estimation.

References

1. https://hgic.clemson.edu/factsheet/tomato-diseases-disorders/
2. Applalanaidu, Majji & Gopalakrishnan, Kumaravelan. (2021). A Review of Machine Learning Approaches in Plant Leaf Disease Detection and Classification. 716-724. 10.1109/ICICV50876.2021.9388488.
3. David, Hepzibah & Ramalakshmi, K. & Gunasekaran, Hemalatha & Ramachandran, Venkatesan. (2021). Literature Review of Disease Detection in Tomato Leaf using Deep Learning Techniques. 274-278. 10.1109/ICACCS51430.2021.9441714.
4. Al-Hiary, Heba & Bani-Ahmad, Sulieman & Ryalat, Mohammad & Braik, Malik & Alrahamneh, Zainab. (2011). Fast and Accurate Detection and Classification of Plant Diseases. International Journal of Computer Applications. 17. 10.5120/2183-2754.
5. Shijie, Jia & Peiyi, Jia & Siping, Hu & Haibo, sLiu. (2017). Automatic detection of tomato diseases and pests based on leaf images. 2537-2510. 10.1109/CAC.2017.8243388.
6. https://www.kaggle.com/datasets/krishrocks/plantvillage
7. https://www.kaggle.com/datasets/rahimanshu/ccmt-plant-disease-dataset
8. S. G. E. Brucal, L. C. M. de Jesus, S. R. Peruda, L. A. Samaniego and E. D. Yong, "Development of Tomato Leaf Disease Detection using YoloV8 Model via RoboFlow 2.0," *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)*, Nara, Japan, 2023, pp. 692-694, doi: 10.1109/GCCE59613.2023.10315251.
9. Priyadharshini, G. et al. “Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN.” *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)* 1 (2023): 1540-1545.
10. Patil, Sanjay & Shrikant, Dr & Bodhe, K. (2011). Leaf disease severity measurement using image processing. International Journal of Engineering and Technology. 3.
11. Yao C, Yang Z, Li P, Liang Y, Fan Y, Luo J, Jiang C, Mu J. Two-Stage Detection Algorithm for Plum Leaf Disease and Severity Assessment Based on Deep Learning. *Agronomy*. 2024; 14(7):1589. https://doi.org/10.3390/agronomy14071589
12. Kwabena, P. M. (2023, Apriil). Dataset for Crop Pest and Disease Detection, Version 1. Retrieved March 19, 2023 from https://data.mendeley.com/datasets/bwh3zbpkpv/1.
13. Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat and Nipun Batra (2020). PlantDoc: A Dataset for Visual Plant Disease Detection, Version 1. Retrieved March 19, 2023 from https://www.kaggle.com/datasets/abdulhasibuddin/plant-doc-dataset.
14. https://www.image-net.org/challenges/LSVRC/2012/
15. https://docs.ultralytics.com/