Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 10(100)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2022.10100136

Leaf disease severity classification with explainable artificial intelligence using transformer networks

Revanasiddappa Bandi[1*], Suma Swamy[2] and Arvind C. S[3]
Assistant Professor, Department of Computer Science, Dr. Ambedkar Institute of Technology, Bengaluru, Karnataka, India, 560056[1]
Professor, Department of Computer Science, Sir. M. Visvesvaraya Institute of Technology, Bengaluru, Karnataka, India, 562157[2]
Senior Researcher, SBIC, Agency for Science, Technology and Research (A*STAR) Singapore, 138632[3]

Received: 12-November-2022; Revised: 27-March-2023; Accepted: 29-March-2023

Abstract
Agribusiness is the main source of income for roughly 70% of people who reside in rural areas. India is the world's second-largest producer of pulses, textile raw materials, spices, coconuts, and other agricultural products. India's gross domestic product (GDP) is significantly impacted by the agriculture industry. Technology advancements help the agricultural industry to forecast various elements, such as soil quality, crop quality, and, disease detection to boost crop yield. Disease detection is one of the essential tasks that have to be carried out in agriculture. The early identification of the leaf disease helps to prevent further spread to other leaves in the plant by which the yield can be improved. In this work, plant leaf disease detection and stage classification are performed based on the severity of leaf infection. A deep learning model, you only look once version5 (YOLOv5) is used to detect plant leaf disease then background of the diseased leaf is removed using U2-Net architecture followed by stage classification performed using vision transformer (ViT) for classifying it as different stages such as low, moderate, and high. A recommendation solution has been provided to mitigate the leaf disease. YOLOv5 was trained using different open-source datasets namely 1) PlantDoc and 2) Plantvillage. This work mainly concentrates on the apple leaf for performing stage classification. The YOLO v5 gives a maximum f1-score of 0.57 at a confidence score of 0.2 and the vision transformer with a background image gives an f1-score of 0.758 and without a background image, 0.908 f1-score is achieved.

Keywords
You only look once version5(YOLOV5), Vision transformer (ViT), Computer-aided disease detection system (CADS), Region proposal network (RPN), Natural language processing (NLP), Explainable artificial intelligence (XAI), Deep convolutional neural network (DCNN).

1.Introduction
The agricultural sector is considered India's most crucial sector. In India, farming is the most common occupation and a significant source of revenue [1]. India is second in the world in terms of population, with 70% of its people living in villages and relying primarily on agriculture for their subsistence. Farmers cultivate a wide variety of crops based on numerous factors such as crop conditions, the environment, soil conditions, local farming practices, new variants of pathogens and various illnesses, etc. Presently farmers are incurring losses due to changes in climatic conditions, plant disease is one of the main challenges to food security since it significantly lowers crop quality and output [2].

Therefore, it has been difficult to diagnose diseases correctly and discover them early. A variety of methods, including soil management, crop rotation, irrigation, genetic modification, harvesting methods, precision farming, weed control, and pest and disease control, are used to boost the yield. So, it is crucial to identify pesticides and diseases early if you want to boost a crop's production. The traditional method of identifying plant diseases is by visual inspection [3]. This can result in incorrect disease identification, and

*Author for correspondence

the use of inexperienced pesticides can have long-term negative effects on the crops and soil.

As a result, there is a need for a computer-aided disease detection system (CADS) [4] powered by artificial intelligence (AI) to assist farmers to improve their crop yield. CADS consists of computer vision, machine learning, and a decision support system that can detect disease using imaging information. CADS systems for plant disease detection using deep learning architectures were developed on high computational resourcing platforms by using open-source datasets such as PlantDoc [5], PlantVillage [6], and also several custom-based plant disease datasets. CADS system with transfer learning methodology was developed to detect black rot, bacterial plaque, and rust diseases using region proposal network (RPN) resulting in improved accuracy compared to the traditional method [7] with high run time complexity.

Deep learning techniques have been used in recent years to accurately classify leaf diseases. However, the lack of interpretability of these models has limited their adoption in practical applications. Explainable artificial intelligence (XAI) aims to provide insight into the decision-making process of AI models, making them more transparent and trustworthy. Transformer networks, a type of deep learning architecture, have been used in various natural language processing (NLP) tasks and have shown promising results in computer vision tasks [8]. Transformers in machine learning is made up of numerous layers of self-attention. The most recent developments in computer vision, which meet state-of-the-art standard accuracy with enhanced parameter efficiency, are only one example of how machine learning advancements show great potential for a general learning technique that can be utilized for a number of data modalities [9].

The main objective of this paper is to accurately identify and classify plant diseases using deep learning networks and to offer appropriate treatments to stop the disease from spreading to other plant leaves. Some researchers used outdated traditional procedures, or they made use of slow, time-consuming features. According to the literature review conducted, the work's motivation is to perform stage classification based on the proportion of disease severity, which was not taken into account in the past. Hence, this issue has been taken into account in this work. This study uses you only look once version 5 (YOLOv5), a deep learning technique, to detect plant diseases and vision transformer (ViT) classifier is used to perform the stage classification. If it is between 0 to 30 then the leaf disease is in the low severity stage, if the severity is between 31 to 60 then the leaf disease is in the moderate stage and if the percentage of severity is 61 to 100 then the leaf disease is in the high severity stage.

The organization of the paper is as follows: section 1 addresses the introduction of agriculture background, challenges of the previous literature, motivation of the work, objectives of the paper, and contribution of the paper. Section 2 explains the literature review carried out on plant disease detection and classification, section 3 discusses the different methods adopted for the leaf disease severity classification, experimental setup for leaf disease detection training, and multistage leaf disease classification, and section 4 describes the results carried out for performance evaluation of leaf disease detection using plantdoc dataset, training and testing analysis of leaf disease detection, performance analysis of background removal technique and multistage leaf disease classification, end-to-end leaf disease detection and classification results for apple rust and apple scab and finally comparative analysis of multistage leaf disease classification with and without background using ViT.

2.Literature review

The major factors that affect leaf disease include changes in climatic conditions such as temperature, humidity, soil moisture, light intensity, PH level, etc., and pathogenic organisms such as viruses, bacteria, fungi, and insects. Several authors discussed how to mitigate leaf disease using different techniques.

The textural statistics application for the diagnosis of plant leaf disease was described by Dhaygude and Kumbhar [10] using various procedures such as transforming an image from red green blue (RGB) to hue saturation value (HSV), hiding and removing green pixels based on their threshold value, 32×32 patch size segmentation is used to separate the relevant segments, and texture statistics are then calculated using the color occurrence technique to assess the existence of disease. The literature review is limited in terms of providing information on the specific vision-based algorithm and neural networks can be used to improve the classification process, as well as any potential challenges or drawbacks associated with their implementation.

Sethy et al. [11] described K-means clustering, multiclass support vector machine (Multiclass-SVM), and particle swarm optimization (PSO) as used to diagnose and detect rice leaf disease. For feature extraction, gray level cooccurrence matrix (GLCM) [12] has been used. PSO increased the detection accuracy to 97.91%. The study's generalizability may be limited as it only collected a small number of specimens with a limited variations of infected rice leaves.

Khirade and Patil [13] proposed the use of image processing and the internet of things to detect plant diseases, specifically on hill banana leaves, by taking images with a camera sensor and transmitting them to cloud storage using the Raspberry Pi3 model. The study used random forest classifier from GLCM features to detect three diseases namely, Black Sigatoka, Bunchy Top virus, and Yellow Sigatoka. This study is limited to the hill banana dataset and they have not performed performance evaluation on the dataset.

Cao et al. [14] suggested a multi-scaled convolutional object detection network. They acquired multi-scaled features using deep convolutional networks and incorporated deformable convolutional structures to counteract geometric alterations. Experiments demonstrate that the proposed framework significantly increases the accuracy of recognizing small target objects with geometric deformation and the speed/accuracy trade-off. This paper can be extended limited to the networks in the field of video object detection.

Guo et al. [15] proposes a deep learning-based mathematical model for plant disease detection and recognition, using a RPN and Chan-Vese algorithm [16] to recognize and localize leaves and segment images based on symptoms. The model outperformed conventional approaches in detecting rust, bacterial plaque, and black rot diseases with an accuracy of 83.57%. The Chan-Vese algorithm's iterative calculation time could be shortened, though. To expedite training and iteration, future research will construct zero initial sets using neural networks. The suggested algorithm has important effects on ecological preservation, sustainable agricultural production, and intelligent agriculture. The Chan-Vese technique requires perpetual iterative calculation and takes a lengthy time, which is not conducive to the method's ability to produce quick identification results.

Hassan et al. [17] explore the use of deep convolution neural network (DCNN) models for the purpose of identifying and diagnosing plant diseases from the leaves. The models were developed using depth separable convolution to cut down on the number of parameters and computational costs. The models were trained using a dataset that included 14 plant species and 38 different disease classes. The outcomes demonstrated high disease classification accuracy rates that outperformed custom-feature-based methods and needed less training time than previous deep learning models. Mobile devices can also be employed with the MobileNetV2 architecture that was used in this study. The paper concludes that DCNN models have promising potential for the efficient identification of plant diseases, including in real-time agricultural systems.

Rashid et al. [18] address the challenges in spotting potato leaf (leaflet) diseases (PLD) in their early phases due to differences in crop category, signs of the disease, and environmental variables. While several machine learning methods have been created to identify potato foliage diseases, they are only applicable in certain areas. In order to identify early blight and late blight diseases, the article suggests a multi-level deep learning model for potato leaf disease identification that makes use of image segmentation and a convolutional neural network. The model beat state-of-the-art models in terms of accuracy and processing cost on a dataset gathered from the Central Punjab area of Pakistan, achieving a 99.75% accuracy rate. The study only concentrates on identifying a specific disease on a leaf, which is a limitation of the article. A limitation of the paper is that the study only focuses on detecting a single disease on a leaf and does not include an assessment of disease location or severity. In addition, the PLD dataset could be improved, and further research is needed to develop an IoT-based monitoring system, website, and mobile application.

YOLOv1 proposed by Redmon et al. [19] uses the Darknet framework. It was trained on the ImageNet-1000 dataset with an input picture size of 224×224 and can identify objects at a rate of 45 frames per second. The limitation is that it cannot generalize the objects if the image has various dimensions or identify the objects accurately when they are tiny.

YOLOv2 and YOLO9000 proposed by Redmon and Farhadi [20] employs Darknet 19 design, which has 19 convolutional layers, 5 max-pooling layers, and a SoftMax layer. It can conduct batch normalization and anchor boxes on input images up to 448×448. Additionally, the mean average precision (mAP) of multi-scale training has increased by 2% by introducing batch normalization to the convolutional layers in the design. On Visual Object Classes Challenge-2007 (VOC 2007), YOLOv2 receives 76.8 mAP at 67 FPS. YOLOv2 achieves 78.6 mAP at 40 FPS, beating cutting-edge techniques like faster region-based convolutional neural networks (Faster R-CNN). Despite having detection data for only 44 of the 200 classes, YOLO9000 receives 19.7 mAP on the ImageNet detection validation set. The algorithm has limited capacity to identify items of different shapes and dimensions. YOLO9000 finds it difficult to display certain products, like eyeglasses.

YOLOv3 was proposed by Redmon and Farhadi [21] and employs the Darknet-53 network, which consists of 53 convolutional layers, as its feature generator. With an input image size of 320×320, it primarily consists of 3x3 and 1x1 filters with shortcut links and uses feature pyramid network (FPN) to make class forecasts. More than 80 distinct items can be recognized by YOLOv3 in a single image. The error rate can be significantly decreased and runs in 22 ms at mAP of 28.2, three times as quickly as single shot detector (SSD) but with the same accuracy. YOLOv3 struggles with little objects that appear in groupings, such as flocks of birds.

YOLOv4 was proposed by Bochkovskiy et al. [22] employs cross-stage partial architecture (CSPDarknet53), a bag of freebies, and a bag of specials as architectural styles. Features like cross-iteration batch normalization (CBN), pan aggregation network (PAN), weighted-residual-connections (WRC), and cross-stage-partial connections (CSP) are applicable to the majority of tasks, models, and datasets. YOLOv4 was rated as one of the best models for speed and accuracy for the Common Objects in Context (COCO) dataset, even if its overall accuracy lagged behind that of EfficientDet largest model. YOLOv4 Struggles to recognize close objects because each grid can suggest only two bounding boxes.

YOLOv5, a creation of Ultralytics open-source research on cutting-edge vision AI techniques, was described in [23] with a 416x416 input image size and features like auto-learning bounding box anchors, 16-bit floating point precision, and new model configuration files. Object detection has been carried out precisely and effectively using .yaml files, cross-stage partial networks as the backbone, preliminary assessment metrics, etc.

Faster R-CNN were discussed by Ren et al. [24] for real-time object detection using RPN. Rapid R-CNN leverages the RPN's fully trained region proposals to make accurate detections. They merged two networks by merging RPN and Faster R-CNN into a single network and sharing their convolutional features. The RPN component gives instructions to the unified network on where to conduct attention-based searches. Using the PASCAL VOC2007, VOC2012 [25, 26], and Microsoft (MS) COCO datasets [27], the proposed approach running on a graphics processing unit (GPU), delivers state-of-the-art object detection accuracy with just 300 proposals per image. The foundation of the top-scoring submissions in a number of tracks in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and COCO 2015 competitions are faster R-CNN and RPN.

Tan and Le [28] proposed a novel scaling method that scales all depth, breadth, and resolution parameters equally using a simple yet immensely powerful compound coefficient. By scaling up Mobile Nets and ResNet, the authors demonstrated the viability of this approach. In order to advance even further, neural architecture search was used to rebuild a fresh basic network, scale it up, and develop the Efficient Nets family of models, which outperforms prior ConvNets in terms of accuracy and productivity. For instance, the proposed EfficientNet-B7 performs better than the most well-known ConvNet since it is 6.1 times quicker and 8.4 times smaller at inference. In addition to obtaining cutting-edge 84.3% top-1 accuracy on ImageNet. The Efficient Net model can be scaled up extremely successfully using ImageNet and five regularly used transfer learning datasets, outperforming state-of-the-art accuracy with orders of magnitude fewer parameters and floating-point operations per second (FLOPS).

Malik et al. [29] explained the sugarcane crop disease detection using deep learning models such as VGG-19, RseNet-34, and Resnet-50. Five sugarcane diseases have been taken into consideration, and images of the situations were taken using cameras at various resolutions and illumination levels. The proposed model was developed using data from the sugarcane industry and demonstrated robustness in identifying complicated patterns by achieving 93.20% accuracy on testing data and 76.40% on data from online sources. The limitation of this study is that, a need for a larger dataset for both tasks, which is believed to be crucial for improving the results.

Toda and Okura [30] described various layer-wise and neuron-wise visualization approaches. Training is carried out by using publicly accessible datasets. This paper demonstrated how neural networks can acquire the texture and color of lesions that are unique to each disease. They deleted the layers from the network that were not helping the development of an effective model, reducing the parameters by 75% without affecting the accuracy. They have used limited training data, overfitting the training data, and lack transparency in the decision-making process of deep neural networks.

Nagasubramanian et al. [31] explained an innovative method, a 3D deep convolutional neural network (DCNN) that instantly incorporates hyperspectral data and the trained model to generate physiologically relevant justifications. The authors concentrated on charcoal rot; a fungal infection transmitted through soil that reduces the production of soybean crops globally.3D DCNN has an infected class F1 score of 0.87 and a classification accuracy of 95.73%, was built on hyperspectral imaging of inoculated and mock-inoculated stem images. In addition to offering excellent accuracy, it also provides physiological insight into model predictions, increasing model predictions' credibility. The study has a potential limitation of using a smaller dataset size, but the model's robustness is tested through fivefold cross-validation.

Liu and Wang [32] described a method for early detection of tomato leaf spots using the MobileNetv2-YOLOv3 model with GIoU bounding box regression loss function. The proposed system shows improvements in detection accuracy and validates real-time detection. The hypothetical results demonstrate high F1 score (94.13%), probability (92.53%), and an average Intersection over Union (IOU) value (89.92%). The system also achieves a high detection speed of 246 frames per GPU and a fast extrapolation speed of 16.9ms for one 416×416 frame. The proposed algorithm should be implemented on a computer or mobile application for practical use in real crop farming so that farmers can easily get support for their crops anywhere and anytime.

Malik et al. [33] explained a hybrid deep learning model which uses a stacked ensemble learning technique to combine MobileNet and visual geometry group-16 (VGG-16) models to generate a composite model. Authors considered four categories of sunflower diseases. They have also created a dataset using Google images and the proposed model outperformed other models in the dataset with 89.2% accuracy. A limitation of the paper is that there are different species of sunflower plants and some diseases can have similar symptoms, making their identification and classification difficult.

Tammina [34] explained transfer learning which is used to address a variety of classification, regression, and clustering-related problems. This work used one of the trained models, VGG-16 with DCNN, to recognize images. The basic convolutional neural network model's validation accuracy is 72.40 percent. The model's accuracy increased to 79.20% with the help of image augmentation. Finally, the VGG-16 pre-trained model, which was created using a sizable image dataset and improved with images, achieved an accuracy of 95.40%.

Sandler et al. [35], described MobileNetV2 which improves the cutting-edge performance of mobile models across a variety of model sizes, workloads, and benchmarks. A novel framework called Single Shot Detector Lite (SSDLite), a mobile DeepLabv3 version of DeepLabv3 that is built on an inverted residual structure with shortcut connections between the thin bottleneck layers is described in this study. It also demonstrates how to build mobile semantic segmentation models. The intermediate expansion layer uses lightweight depth-wise convolutions as a source of non-linearity to filter features. When used with the ImageNet dataset, the suggested architecture advances the state of the art for a number of performance measures. Our network outperforms state-of-the-art Realtime detectors for the job of object identification on the COCO dataset in terms of accuracy and model complexity. Interestingly, when combined with the SSDLite detection module, the proposed solution uses 20 fewer computations and 10 fewer parameters than YOLOv2.

DenseNet and EfficientNet, Deep convolutional neural network models, were described by Srinidhi et al. [36] and used to successfully detect four kinds of apple plant diseases from images of apple plant leaves (leaflets). The different categories are scab, rust, healthy, and several diseases. The dataset for apple leaf disease is improved in this work using data augmentation and image annotation techniques such as Canny Edge Detection, Blurring, and Flipping. Based on an improved dataset, models using

EfficientNetB7 and DenseNet are recommended, achieving an accuracy of 99.8% and 99.75%, respectively, and resolving convolutional neural network shortcomings. Models may become even more accurate and dependable by using powerful validation techniques.

Sibiya and Sumbwanyambe [37] introduced the benefits of fuzzy logic principles. The current study aims to update the algorithm used in the Leaf Doctor application, which is used to determine the severity of plant leaf diseases. This strategy will develop the technology of precision agriculture by introducing an algorithm that might be used in smartphone programs like the Leaf Doctor app. Applications developed using the technique presented in this paper will help beginner users and non-plant pathologists grasp the estimated severity of the disease. For smartphone apps that don't employ fuzzy logic, this study required less time and is recommended in order to acquire relevant findings. The recommended approach for color threshold segmentation was tested using a Fiji image. The fuzzy logic inference system was modeled and tested using LabVIEW software [38]. It is indicated that Leaf Doctor is one program that, in subsequent upgrades, will employ the suggested methodology.

Shi et al. [39] outlined different studies on convolution neural network (CNN) based plant disease severity assessment in terms of classical CNN frameworks, improved CNN architectures, and CNN based segmentation networks, depending on the network architecture. The study also provided a detailed comparative analysis of the advantages and disadvantages of each approach. Moreover, the study investigated common methods for acquiring datasets and performance evaluation metrics for CNN models. Overall, this literature survey provides insights into the current state of research on the use of deep learning techniques for plant disease severity assessment and highlights potential areas for future research.

Abd et al. [40] introduced a new deep learning technique called ant colony optimization with convolution neural network (ACO-CNN) for disease detection and classification in plant leaves. The technique uses ant colony optimization to enhance the accuracy of disease diagnosis. The proposed method subtracts geometries of color, texture, and leaf arrangement from the provided images using a CNN classifier. The study uses several effectiveness metrics to compare the proposed approach with existing techniques and demonstrates that the proposed approach outperforms existing methods with improved accuracy rates. The study also presents the steps involved in disease detection, including image acquisition, image separation, noise removal, and classification. However, the paper does not consider the stage classification of plant disease.

Mohammed and Yusoff [41] provides an overview of various techniques available for achieving success in machine learning, deep learning, and image processing. It emphasizes the importance of training and testing models with more datasets to increase the accuracy of recognition rates. The review also highlights the need for new and improved deep learning methods to classify plant diseases. While several methods have been used in the past, neural networks like CNN appear to be the best technique due to their flexibility and feature extractor property. Unlike previous models like naive bayes and support vector machine (SVM), CNN can learn additional features from images to provide better output. Therefore, CNN is the best choice for research work in image processing, machine learning, and deep learning due to its ability to learn and extract features from images for reliable output.

The literature review discussed above has demonstrated that plant disease diagnosis is being solved using various image processing, machine learning and deep-learning techniques. However, most of the research work carried out either talks about the detection or classification of plant diseases and few have used image processing methods for stage identification which does not provide a generalized solution for deployment. Hence there is a need for a robust and novel pipeline for plant disease stage classification. In this research work, an end-to-end deployable solution is developed that does leaf detection using YOLOv5 architecture and ViT architecture [42] that does stage classification. The recommendation engine provides a mitigation solution based on the severity of the disease. Ground truth for stage classification was done by an expert plant pathologist. To reduce time and burden, an unsupervised clustering-based ground-truth generation methodology was developed. Also, a comparative study of F1 score for multistage leaf disease classification with and without background using ViT classifier and its interpretability on disease stage classification was performed on an apple leaf disease.

3.Methods

The perception of the end-to-end multistage leaf disease detection and classification system is shown in *Figure 1*. Multi-stage pipeline consists of six different stages. (i) Data acquisition from PlantDoc and Plantvillage datasets (ii) Data preprocessing was performed using resize, augmentation, and contrast enhancement techniques. (iii) Disease detection model was trained using different parameters and hyperparameter tuning (iv) Background removal was performed to accurately classify the image (v) Stage classification was accomplished based on the severity of leaf infection. (vi) recommendation solution was provided to mitigate the leaf disease. In this research work, apple leaf data was used for multistage classification. The ground truth for different disease stages. (i) low (ii) moderate and (iii) high, were generated using the mean-shift clustering technique and were validated by experts.

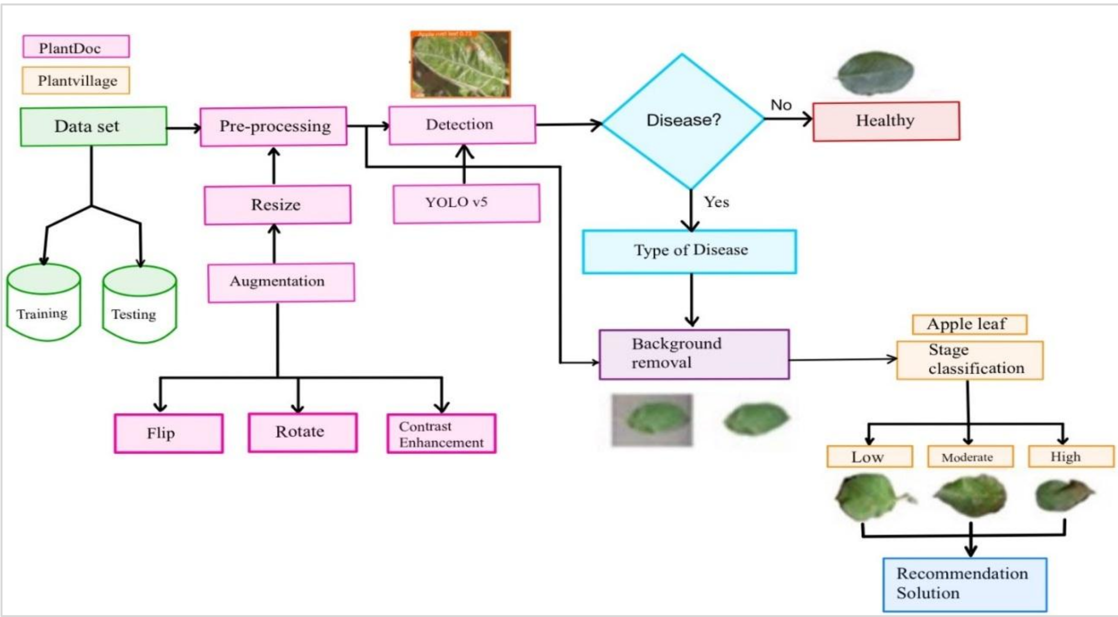


Figure 1 The perception of multistage plant leaf disease detection and classification

3.1Data Acquisition

In this research work, two open-source datasets were used for plant leaf disease detection and classification.

Plantdoc: A dataset for visual plant disease detection: PlantDoc dataset [4] contains cropped leaf images used for standardizing classification models. The researchers used this dataset for training image classification models, VGG-16 InceptionV3 and object identification models, MobileNet, and Faster-RCNN. The dataset can be used to improve general agricultural computer vision tasks, health crop categorization, and plant disease classification. The PlantDoc dataset contains 2572 images which include 13 species and 27 classes out of which 17 classes are diseased and 10 classes are healthy. The different classes and number of images in a particular class are depicted in *Table 1*. These images are used for training the YOLOv5 model to perform leaf region detection and its disease type.

The two sample classes of apple leaf disease i.e., apple rust leaf and apple scab leaf are depicted in *Figure 2*.

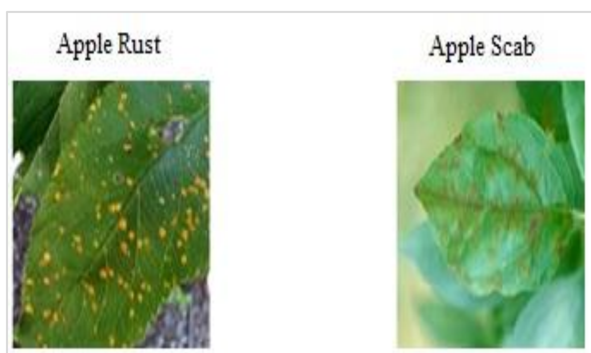


Figure 2 Apple leaf diseases from the PlantDoc dataset

Table 1 PlantDoc dataset used for leaf disease detection

Attributes/Classes	**Ranges/No. of Images**
Tomato two spotted spider mites leaf(leaflet)	2
Tomato mosaic virus leaf(leaflet)	54
Cherry leaf(leaflet)	57
Bell pepper leaf(leaflet)	61
Tomato leaf(leaflet)	63
Grape leaf(leaflet) black rot	64
Soyabean leaf(leaflet)	65
Corn Gray leaf(leaflet) spot	68
Grape leaf(leaflet)	69
Bell pepper leaf(leaflet) spot	71
Tomato yellow virus leaf(leaflet)	76
Apple rust leaf(leaflet)	88
Tomato early blight leaf(leaflet)	88
Apple Leaf(leaflet)	91
Tomato leaf(leaflet) mold	91
Apple Scab Leaf(leaflet)	93
Strawberry leaf(leaflet)	96
Potato leaf(leaflet) late blight	105
Tomato bacterial spot leaf(leaflet)	110
Peach leaf(leaflet)	111
Tomato late blight leaf(leaflet)	111
Blueberry leaf(leaflet)	115
Corn rust leaf(leaflet)	116
Potato leaf(leaflet) early blight	116
Raspberry leaf(leaflet)	119
Squash Powdery mildew leaf(leaflet)	130
Tomato septoria leaf(leaflet) spot	151
Corn leaf(leaflet) blight	191
Healthy	847
Unhealthy	1725
Total	2572

PlantVillage: Dataset of diseased plant leaf images and corresponding labels: The PlantVillage dataset [5] contains 54,309 carefully labeled images on both healthy and diseased plant leaves of 14 different crops.

These data mark the start of a continuous crowdsourcing effort to enable computer vision methods to assist in resolving the issue of crop plant yield losses caused by viral diseases. This dataset is used for plant disease detection systems and this is a popular dataset that is used by many researchers. In this work samples of the datasets were used for stage classification using ViT. Typical images of apple rust and apple scab from the plantvillage dataset are shown in *Figure 3.*

3.2Data Pre-processing

Before the model training, resize the images to 416×416 size as it is a prerequisite for YOLOv5 architecture [43]. To avoid overfitting and to create generalization, data augmentation techniques have been applied like (i) flip (ii) rotate, and (iii) contrast enhancement.

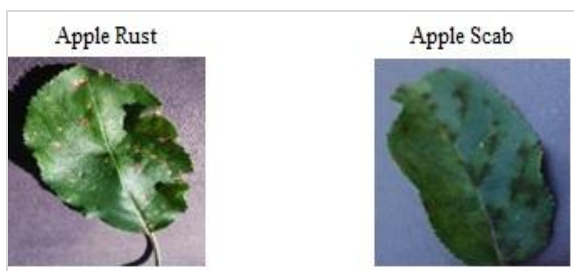


Figure 3 Apple leaf diseases from the plantvillage dataset

To train the YOLOv5 model, the dataset is split into two parts 90% for training and 10% for testing the model. *Table 2* shows the number of training and testing images considered for the YOLOv5 model training of each class disease of PlantDoc.

Table 2 Training and testing data for training leaf disease detection using plantDoc dataset

Class name	**Training images**	**Testing images**
Tomato two spotted spider mites leaf(leaflet)	2	0
Tomato mosaic virus leaf(leaflet)	44	10
Cherry leaf(leaflet)	47	10
Bell pepper leaf(leaflet)	53	8
Tomato leaf(leaflet)	55	8
Grape leaf(leaflet) black rot	56	8
Grape leaf(leaflet)	57	12
Soyabean leaf(leaflet)	57	8
Bell pepper leaf(leaflet) spot	62	9
Corn Gray leaf(leaflet) spot	64	4
Tomato yellow virus leaf(leaflet)	70	6
Apple rust leaf(leaflet)	78	10
Tomato early blight leaf(leaflet)	79	9
Apple Leaf(leaflet)	82	9
Apple Scab Leaf(leaflet)	83	10
Tomato leaf(leaflet) mold	85	6
Strawberry leaf(leaflet)	88	8
Potato leaf(leaflet) late blight	97	8
Tomato bacterial spot leaf(leaflet)	101	9
Tomato late blight leaf(leaflet)	101	10
Peach leaf(leaflet)	102	9
Blueberry leaf(leaflet)	104	11
Corn rust leaf(leaflet)	106	10
Potato leaf(leaflet) early blight	108	8
Raspberry leaf(leaflet)	112	7
Squash Powdery mildew leaf(leaflet)	124	6
Tomato septoria leaf(leaflet) spot	140	11
Corn leaf(leaflet) blight	179	12
Healthy	757	90
Unhealthy	1579	146
Total	2336	236

3.3Leaf disease detection

Leaf disease is detected using one of the most advanced object identification techniques YOLOv5. The YOLOv5 uses a single neural network to process the entire image, then separates it into parts along with the bounding box probabilities of each component. The network takes an image as input and produces a set of bounding boxes and class probabilities as output. The bounding boxes represent the location of the objects in the image, while the class probabilities represent the likelihood of each object belonging to a specific class. One of the main improvements of YOLOv5 is its architecture. It uses a scaled approach, where the network is scaled up or down depending on the size of the input image. This allows YOLOv5 to be more accurate and faster than previous versions of YOLO. There are two choices for training the YOLOv5 model. a) Activation and Optimization Function: Leaky rectified linear unit (ReLU) and Sigmoid are used as activation functions. Stochastic gradient descent (SGD), and adaptive moment estimation (ADAM) are used as optimizer functions. In YOLOv5, the final detection layer uses the sigmoid activation function whereas the middle/hidden layers use the Leaky ReLU activation function. b) Loss Function: A compound loss is calculated for the YOLO based on the objectness score, class probability score, and bounding box regression score. For the loss computation of class probability and object score, Ultralytics employed the Binary Cross-Entropy with Logits Loss function from the PyTorch library. The loss function of YOLO is described in Equation 1.

$$loss = l_{box} + l_{cls} + l_{obj} \tag{1}$$

where lbox is a bounding box regression function, lcls is a loss function of classification and lobj is a loss function of confidence. The regression loss function of the bounding box is written as in Equation 2.

$$l_{box} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i.j}^{obj} b_j (2 - w_i X\, h_i)\, [(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2 + (w_i - \widehat{w}_i^j)^2 (h_i - \hat{h}_i^j)^2] \tag{2}$$

The classification loss function is written as in Equation 3.

$$l_{cls}=\lambda_{class} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{obj} \sum_{c\in classes} p(c)\log(\hat{p}i(c)) \quad (3)$$

The confidence loss function is written as in Equation 4.

$$l_{obj} = \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{noobj} (c_i - \hat{c}_j)^2 + \lambda_{obj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{i,j}^{obj} (c_i - \hat{c}_j)^2 \quad (4)$$

where $\hat{x},\hat{y}$ is the target's real central coordinate, $\hat{w},\hat{h}$ is the target width and height, and λ_{coord} is the position loss coefficient, λ_{class} is the category loss coefficient. If the anchor box at (i, j) contains targets then the value of $I_{i,j}^{obj}$ is 1 otherwise it is 0. $Pi(c)$ is the target's category probability and $\hat{p}i(c)$ is the category's actual value. the total number of categories C is represented by the length of the two. The input image is passed to the YOLOv5 model which processes it and outputs the detected image with a confidence score as shown in *Figure 4.*

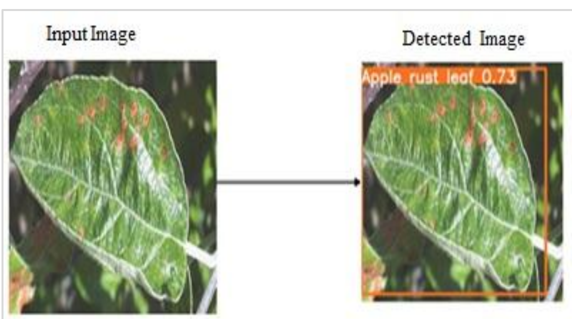


Figure 4 Leaf disease detection using YOLOv5

3.4Multistage infection ground truth preparation

Algorithm 1 explains the steps taken to prepare the multistage infection before ground truth preparation. Input to the algorithm will be apple leaf images from the plantvillage dataset and output will be three different stages of severity of the apple leaf disease. In the first step, the mean-shift clustering technique is applied to the input images to get the region of clusters. In step 2 canny-edge detection method is applied to get the leaf and edges of the infected region. In step 3 calculate the contour to know the area of the diseased part in the leaf. In step 4 different metrics are calculated such as leaf infection area, total area, perimeter, and percentage of the infection on the leaf finally images are grouped into three categories low, moderate, and high.

Algorithm 1: Multi-Stage Infection Ground Truth Preparation Algorithm

Input: Apple Leaf Disease Images from PlantVillage Dataset

Output: Different stage identification as ground truth of apple leaf disease dataset for classification

Step 1: Apply mean-shift clustering to get disease region clusters [44].

Step 2: Canny-edge detection is applied to get leaf and infect region edges [45].

Step 3: Find the contour of the leaf to know the area of the diseased part in the leaf.

Step 4: Leaf infection area, total area, and perimeter are calculated, as the percentage of the infection on the leaf.

Step 5: Based on **Step 4** three different severity categories are grouped as, (i) low (ii) moderate (iii) high severity images.

3.5Background removal

A digital image processing technique called background removal can be used to separate a picture's components into interesting and undesirable areas. Before further analysis and processing, background reduction is necessary for many applications of image processing and computer vision. To reduce false classification results, the background removal technique is adopted using U2-Net architecture [46]. It is a two-level nested U-structure, consisting of 11 stages on the top and each stage is configured by a residual U-block (RSU) on the bottom as shown in *Figure 5(a).* Background removal was performed on the apple leaf using U2-Net architecture where the RGB image of the apple leaf is converted into the binary image and then passed on to the U2-Net model which removes the background of the binary image using residual U-blocks. *Figure 5(b)* shows the removal of background noise from the leaf image which is not required for the classification and it will affect the accuracy of the classifying image. Initially, the input image is passed to the U2-Net model, it converts to a binary image then background noise is removed for the binary image, after that background noise is removed for the RGB image to get the desired output image.

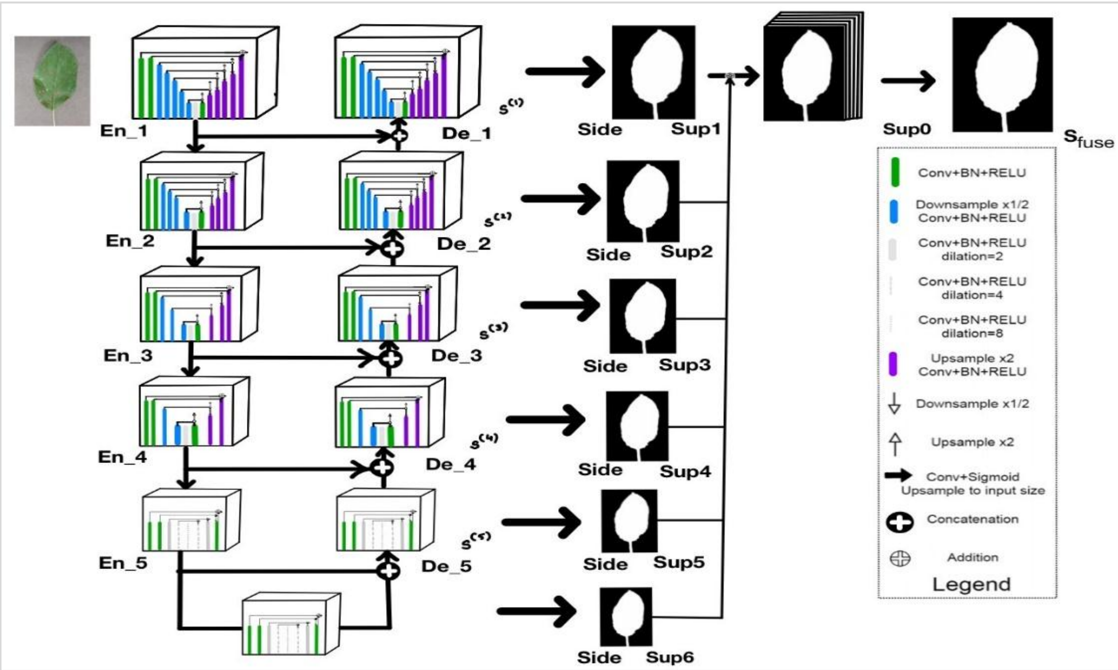


Figure 5(a) Illustration 0f U2-Net architecture

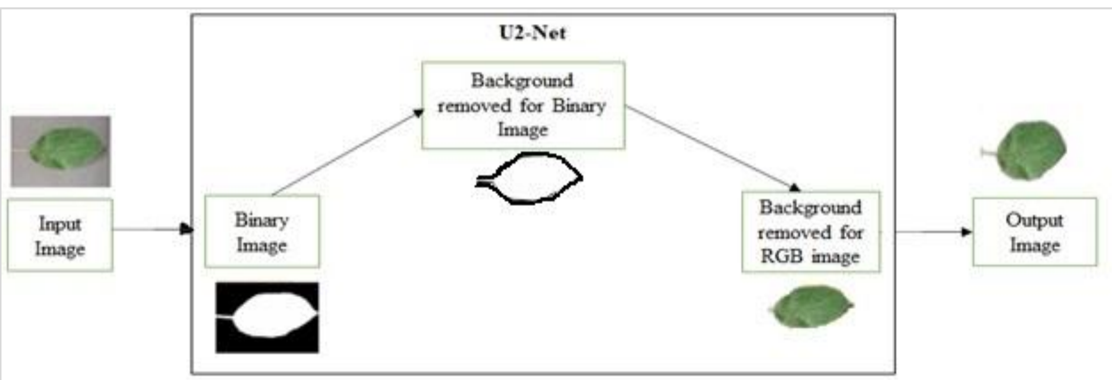


Figure 5(b) Plant leaf image background removal using U2-Net architecture

3.6Multistage classification

The proposed multi-stage classification is performed by using deep learning architecture ViT. It is an image classification technique and it employs a transformer-like architecture that gives patches over the images. A flattened vector of pixel values from patches of size 16×16 makes up the input sequence, and each of these patches is embedded linearly. These embedded patches are added up and the result is passed to the standard transformer encoder as shown in *Figure 6*. The ViT is the intersection between NLP and computer vision. It has achieved high performance for object detection, classification of images, computer vision applications, and semantic segmentation. Transformers applied directly on image patches that are in sequence can work well on image classification tasks.

Algorithm2 explains about steps taken for inference leaf disease detection and stage classification. Input to the algorithm is a plant image from the Plantdoc dataset and output is detected disease and their severity. In step 1 the input image is passed to the leaf disease detection model using YOLOv5 to detect the image. In step 2 check the image for healthy or diseased. If it is diseased find the region of interest (ROI) and perform the background removal on the input image in step 3 and step 4 respectively and finally apply a vision transformer on the background removed image to get the severity of the disease.

Algorithm 2: Inference Leaf Disease Detection and Stage Classification

Input: Plant leaf Image $I_{(x,y)}$ from PlantDoc dataset.
Output: Detected Disease $I_{D(x,y)}$, Severity of Disease $SI_{D(x,y)}$
$P_D \rightarrow$ Plant Disease type
$I_{ROI(X,Y)} \rightarrow$ Region of Interest of Diseased Image
BR()$\rightarrow$ Background Removed Image
ViT()$\rightarrow$ Vision Transformer
$I_{ROI\ BR(x,y)} \rightarrow$ Region of Interest of Background Removed Image
Step 1: Input the image I(x,y) to disease detection model using YOLOv5 to detect the image
Step 2: Check the image for healthy or diseased
Step 3: If it is diseased, find the region of interest of Diseased Image $I_{ROI\ (X,Y)}$
Step 4: Perform background removal on $I_{ROI(X,Y)}$
Step 5: Apply VIT on ROI of background removed image to get the severity of disease $SI_{D(x,y)}$

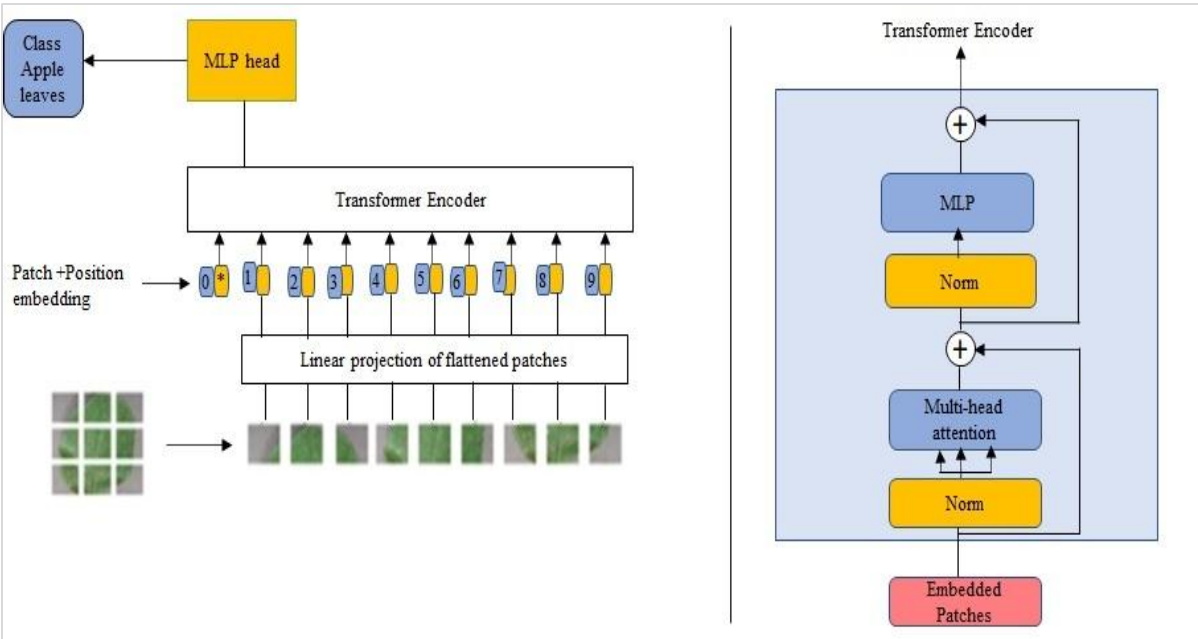


Figure 6 Multistage classification using vision transformer

3.7Experimental setup

The proposed research work was conducted using open-source google collab with K80 NVIDIA GPU. Pytorch library with python 3.7. The Hardware requirements of the proposed work include an Intel Core i7-12th generation system, 1TB Hard Disk, 32GB RAM, and Software resources such as Windows 10 Operating System, google collab with K80 NVIDIA GPU as Software Tool, Python 3.7 Coding Language and Pytorch Library.

3.7.1Training parameters and hyperparameters

The input image size for YOLOv5 should be of size 416×416 dimension and a batch size of 50 i.e., amount of training data in one iteration, 715 epochs are used to train the model, with learning rates of 0.1 and 0.01 to learn the model, and 0.0005 of weight decay is applied. Different training parameters and hyperparameters for training YOLOv5 using the PlantDoc dataset are shown in *Table 3.*

Table 3 Training parameters/hyperparameters for leaf disease detection using YOLOv5

S. No	Training parameters		Values
1	Image size		416 × 416
2	Batch size		50
3	Epochs		715
4	Learning rate	lr 0	0.01
		lr f	0.1
5	Weight decay		0.0005

3.7.2Multistage leaf disease classification

For finding the severity of the plant leaf disease, multistage classification is performed on the plantvillage dataset. The different training parameters used for the plantvillage dataset for classifying the stage of the disease using ViT are shown in *Table 4*. The input image size for ViT should be of size 224×224 dimension and batch size is 64 and a total number of 200 epochs are taken to train the model, learning rate of 2e-5 is used to learn the model and

0.7 of gamma is applied to check how close the training reaches and seed value of 42 has been used. The apple leaf disease class has been chosen for training and testing the data using ViT classifier by considering with background and without background removed images. *Table 5* and *6* shows the statistics of the data used for training, testing, and validation and the number of images used should be equal for the ViT classifier. Different stages of apple scab and rust leaf are low, moderate, and high. The data imbalance problem is faced in this stage because the model requires an equal number of images for training, testing, and validation. So, here we have used data augmentation techniques like rotation, flip and rotate at different angles to maintain an equal number of images for each step.

Table 4 Training parameters/ hyperparameters for leaf disease classification using ViT

S. No.	**Training parameters**	**Values**
1	Image size	224 x 224
2	Batch size	64
3	Epochs	200
4	Learning rate	2e-5
5	Gamma	0.7
6	Seed	42

Table 5 Training, testing, and validation data for multistage leaf disease classification using plantvillage dataset with background images

Type of disease class	**Stage of the disease**	**Train**	**Test**	**Validation**
Apple Scab Leaf	Apple Scab Low	217	217	217
	Apple Scab Moderate	216	216	216
	Apple Scab High	197	197	197
Apple Rust Leaf	Apple Rust Low	216	216	216
	Apple Rust Moderate	203	203	203
	Apple Rust High	211	211	211
Total Images		1260	1260	1260

Table 6 Training, testing, and validation data for multistage leaf disease classification using plantvillage dataset without background images

Type of disease class	**Stage of the disease**	**Train**	**Test**	**Validation**
Apple Scab Leaf	Apple Scab Low	221	221	221
	Apple Scab Moderate	199	199	199
	Apple Scab High	202	202	202
Apple Rust Leaf	Apple Rust Low	187	187	187
	Apple Rust Moderate	227	227	227
	Apple Rust High	216	216	216
Total Images		1252	1252	1252

4.Results

The main goal of the research work is to find the best state of the model for plant leaf disease detection and perform multi-stage classification based on severity level. Experimental results show that the training is done using both original datasets as well as supplemented datasets. So, this model can be installed for real-time prediction.

4.1Performance evaluation

The trained model performance is evaluated by using a confusion matrix as depicted in *Figure 7*. The calculation is performed based on the actual value v/s predicted value that belongs to different classes. In leaf disease detection training using YOLOv5 the mAP values generated for the classes apple scab, apple leaf (leaflet), and apple rust are 0.551, 0.258, and 0.245 respectively at a confidence score of 0.7.

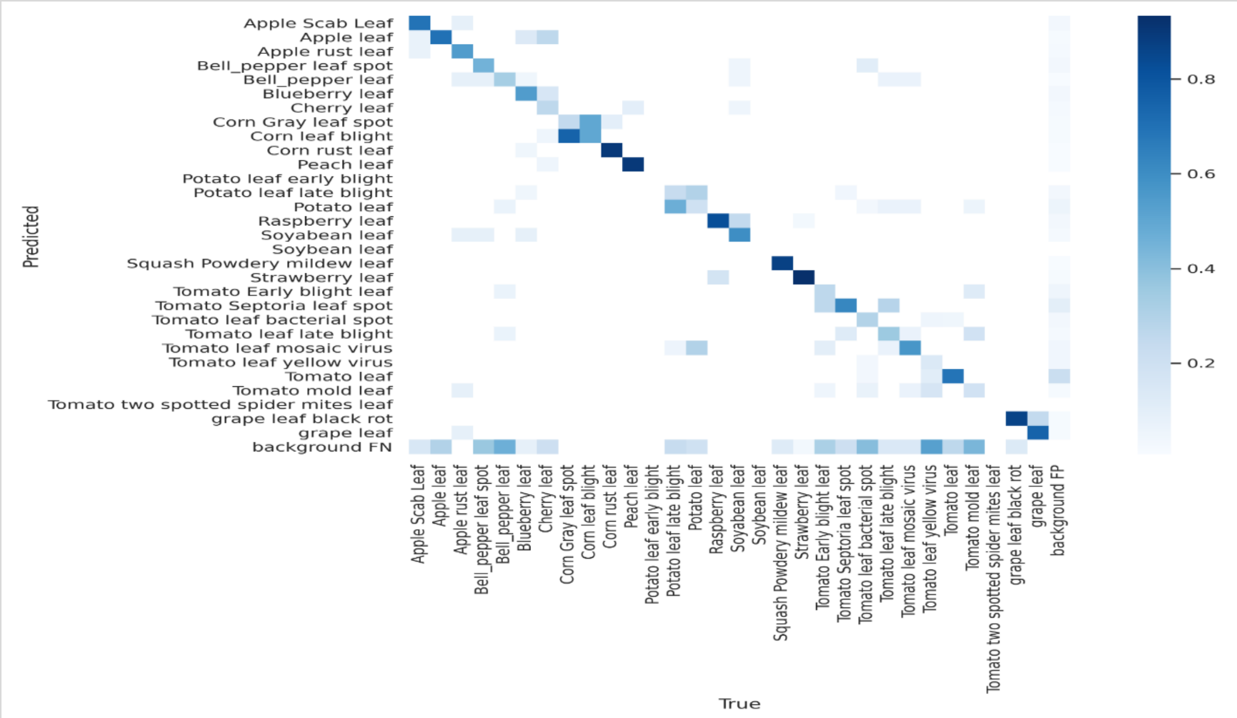


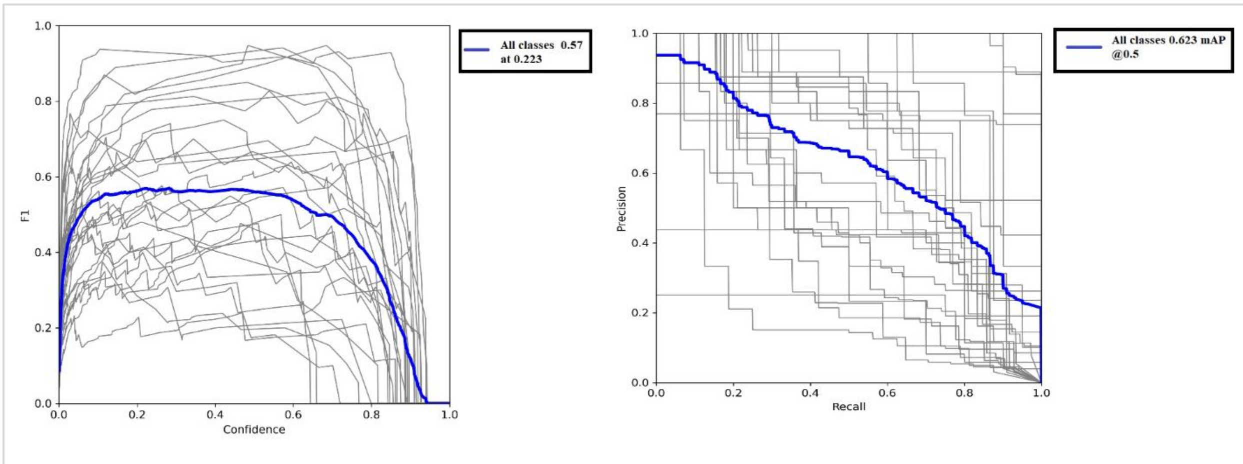
Figure 7 Confusion matrix of leaf disease detection using plantdoc dataset

(a)Leaf disease detection training performance

F1 score is used as a statistical measure to rate performance. The F1 score is applicable for any point on the receiver operating characteristic (ROC) curve. For the F1 score to be high, both precision and recall values should be high. *Figure 8(a)* shows the F1 curve of leaf disease detection training which consists of 27 different classes of the PlantDoc dataset the confidence score is plotted on the x-axis and the F1 score on the y-axis. The grey color in the graph indicates different classes of F1 scores and the blue color indicates the overall F1 score of all the classes. The maximum F1 score is 0.57 at a 0.223 confidence score. A precision and recall curve are a plot of a graph which are drawn based on recall at the x-axis and precision at the y-axis. It is also called a Precision-Recall (PR) curve and is used for different probability thresholds. PR curve uses class imbalance which is estimated on the baseline to inform how well the model shall perform, for the given specific imbalance. The grey color in the graph indicates different classes' PR values and the blue color indicates the overall PR value of all the classes. It is observed that the graph has a descending trend which means if the threshold is lower than you will have more false positive predictions and if the threshold is high then you will have more false negative predictions. The PR-score is 0.623 at a mAP of 0.5 as shown in *Figure 8(b).*

b) Accuracy and Loss graph

In order to better understand the performance of leaf disease detection using YOLOv5, find the different losses such as classification loss(cls_loss), objectness loss (obj_loss) is the confidence of object presence which is 0.028874, and bounding box regression loss (box_loss) is 0.03059 as shown in *Table 7*. Since the dataset contains multiple classes the classification error is 0.006629. Precision measures the percentage of accurate bbox forecasts and how much of the true bbox was properly predicted is measured by a recall, the precision and recall values are 0.66002 and 0.54152 respectively. A mAP is an evaluation metric used in the object detection model and it is calculated by fixing the confidence threshold. mAP 0.5 refers to the mAP at a threshold of 0.5 for IoU which is 0.61814. The average mAP over various IoU thresholds, from 0.5 to 0.95, is represented by the notation mAP 0.5:0.95 which is 0.46136. The validation or testing box_loss(error), obj_loss(error), and cls_loss(error) is 0.031752, 0.011021, and 0.041854 respectively. The same values have been represented using different graphs as training loss, accuracy metrics, and testing or validation loss as shown in *Figure 9.*



a) F1 curve for PlantDoc Dataset b) PR curve for PlantDoc

Figure 8 F1 curve and PR curve of leaf disease detection training

Table 7 Accuracy and loss values of leaf disease detection

Accuracy/Loss	Value
Training/box_loss(error)	0.028874
Training/obj_loss(error)	0.03059
Training/cls_loss(error)	0.006629
Metrics/precision	0.66002
Metrics/recall	0.54152
Metrics/mAP_0.5	0.61814
Metrics/mAP_0.5:0.95	0.46136
Validation/box_loss(error)	0.031752
Validation/obj_loss(error)	0.011021
Validation/cls_loss(error)	0.041854

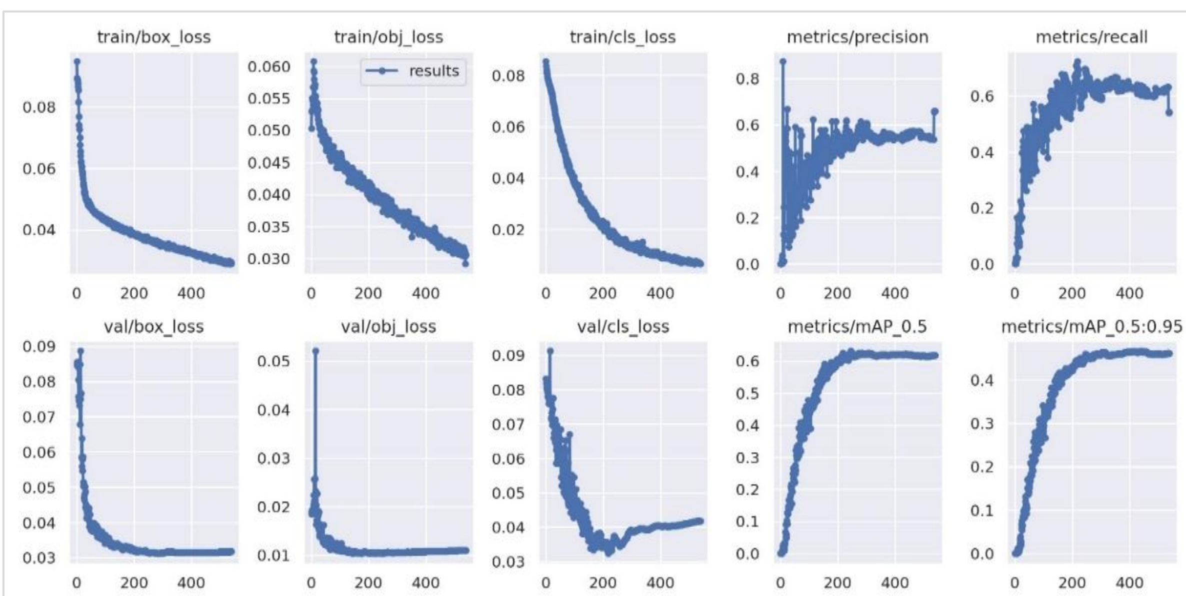


Figure 9 Accuracy and loss graph of leaf disease detection training

c) Leaf disease detection testing analysis

The leaf disease detection testing analysis can be shown using the boxplot which gives confidence scores representation at threshold 0.3, 0.5, and 0.7 on the x-axis and mAP on the y-axis as shown in *Figure 10*. The confidence score or classification threshold is nothing but how confident the machine learning model is that the appropriate intent was assigned. Based on how the neural networks function, the score can range from 0 to 1. Typically, a score for each user input is calculated for each intent, and the highest score is returned as the outcome. It is observed that the best Average mAP of 0.443 at a confidence score of 0.7 as there are no outliers and inter quartile range (IQR) is high. *Table 8* shows confidence score values and corresponding average mAP at the different thresholds. The box confidence score is nothing but a minimum confidence threshold, the model has detected an object.

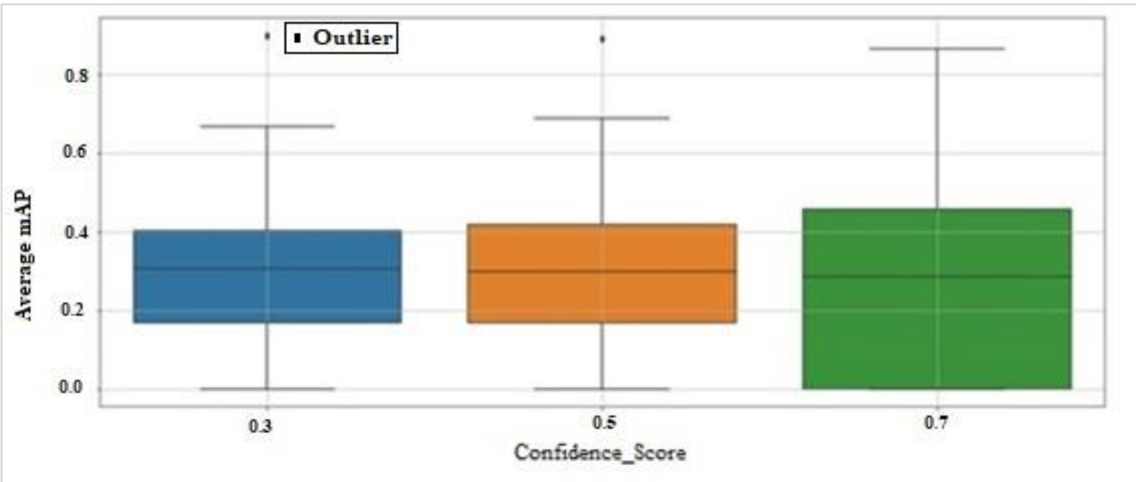


Figure 10 Box plot of leaf disease detection testing on plantdoc dataset

Table 8 Inter quartile range of average map for different confidence scores of leaf disease detection testing

S. No.	Confidence score	Average mAP
1	0.3	0.2345
2	0.5	0.2593
3	0.7	0.443

4.2Performance analysis of background removal technique

To evaluate the performance of background removal manually, two classes of apple diseased leaves are considered, apple scab and apple rust for finding a background of the image using U2-Net architecture. To know the performance of U2-Net, the input image was annotated using an open-source labeling tool called LabelMe [47] to get the binary image as shown in *Figure 11*. The dice score or dice coefficient is used for determining the background removal performance which is a statistical tool used to determine how similar two samples of data are. In this study, the dice score for two classes of apple diseased leaves is calculated using the sum of the intersection of both images to the sum of two images. The dice score of apple scab is 0.750+- 0.097 and apple rust is 0.756 +- 0.093 depicted in the box plot as shown in *Figure 12.*

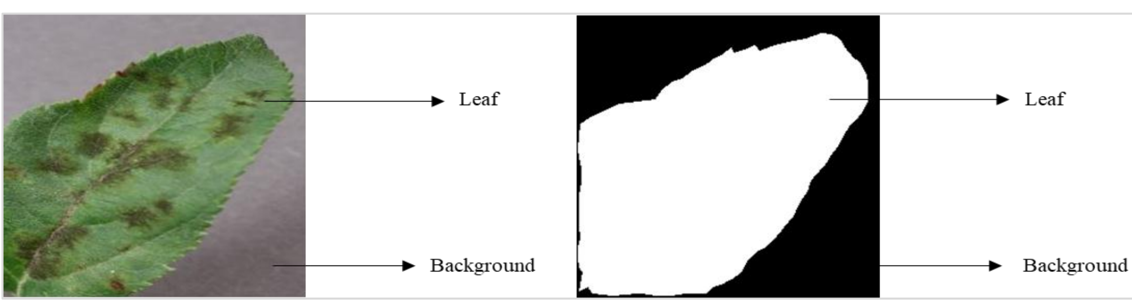


Figure 11 Apple scab leaf RGB image and binary image with background

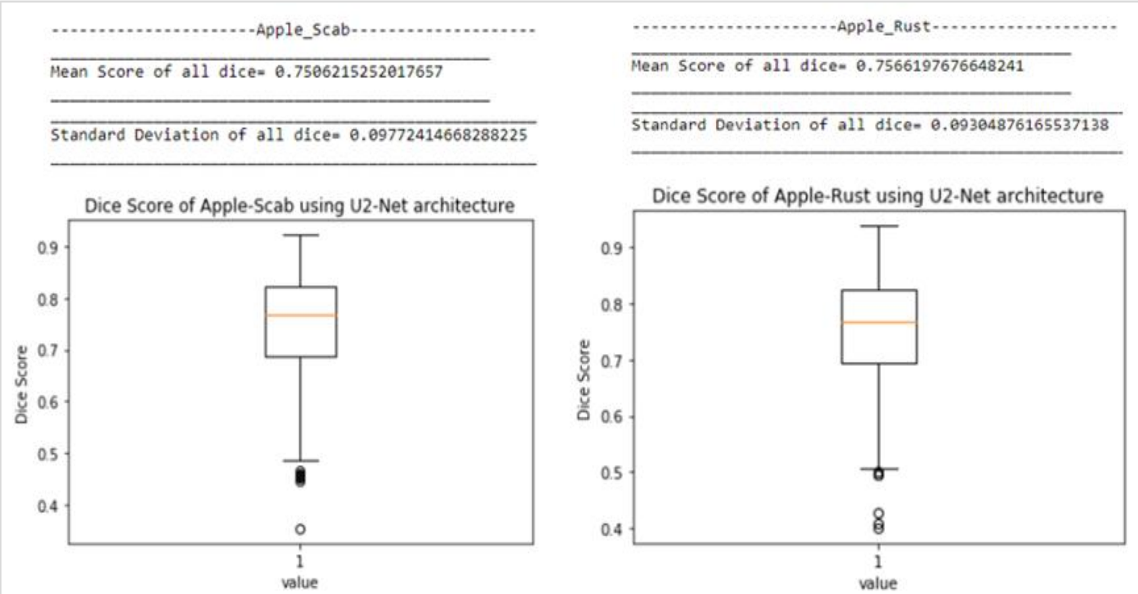


Figure 12 Dice score of apple scab and apple rust using U2-Net

4.3 Performance analysis of multistage leaf disease classification

Classification metrics are employed to assess a classification model, they indicate if the classification is good or unsatisfactory. The various measures include, 1) Accuracy: It is calculated as the proportion of accurate forecasts to all predictions. 2) Confusion matrix: It is a table that contrasts model predictions with the ground truth. 3) Precision: is measured by how many accurate forecasts are produced 4) Recall: is a metric for how many out of all the positive cases in the data that the classifier correctly predicted. 5) F1 Score: the weighted average of Recall and Precision. Therefore, this score takes both false positives and false negatives into account represented in Equation 5.

$$\text{F1 Score} = 2 \times \frac{Recall*Precision}{Recall+Precision} \qquad (5)$$

The performance of multistage leaf disease classification is calculated through the above measures. The output of a classification algorithm is shown and summarized in a confusion matrix as shown in *Figure 13.* The overall F1 score of multistage leaf disease classification with and without background are 0.75842 and 0.90846 as depicted in *Figure 14.*

4.4 Integration of leaf disease detection and severity classification

For the integration model, an input image is passed then it will detect and identify the type of image as apple rust or apple scab then the contour of the image will be shown, A contour image is a visualization of an object where the edges or boundaries of the objects or surfaces are highlighted using lines or curves of a specific color. In other words, a contour image represents the shape and structure of an object or surface by showing its outermost boundaries. The contours are a useful tool for both form analysis and object detection and recognition. Finally, the heat map image is produced, it is a graphical representation of data that uses color coding to visualize the intensity or density of values in a two-dimensional matrix or table. In a heat map, each cell of the matrix is assigned a color that represents the value it contains, with colors ranging from low to high intensity. The colors can be any gradient, but typically, red or yellow colors are used to represent high values, while green or blue colors are used to represent low values. The leaf severity is calculated based on the percentage of infection on the particular leaf using the ViT classifier. In this work, we have tested our model for two classes of apple leaf disease namely apple rust and apple scab. The end-to-end pipeline of the leaf disease detection and classification results for two samples of the apple rust leaf and apple scab with severity and recommendation solutions are shown in *Figure 15* and *Figure 16.*

Figure 13 Confusion matrix of multistage leaf disease classification with and without background

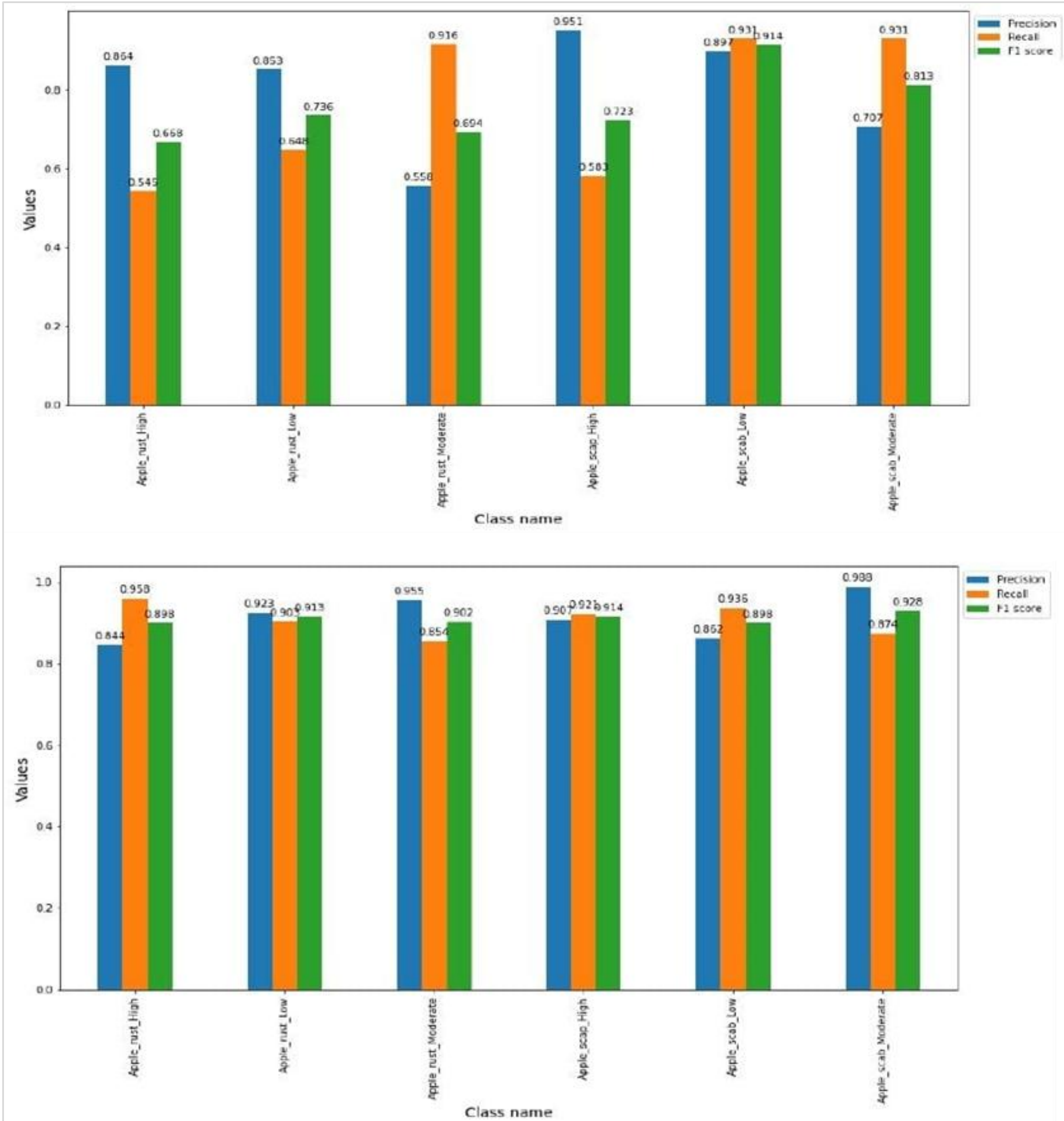


Figure 14 Precision, recall, and F1 score of multistage leaf disease classification with and without background

4.5Comparative analysis

This section addresses the performance of two classes of apple leaf diseases i.e. apple rust and apple scab with their severity levels as low, moderate, and high. The performance standards precision, recall and F1 score have been calculated for each of the severity levels with the support value being considered. *Table 9* and *Table 10* depicts the performance values of multistage leaf disease classification with background and without background. The complete F1 score of classification with background is 0.758, and without background is 0.910. *Table 11* shows the comparison results of the F1 score for multistage leaf disease classification with and without background using ViT.

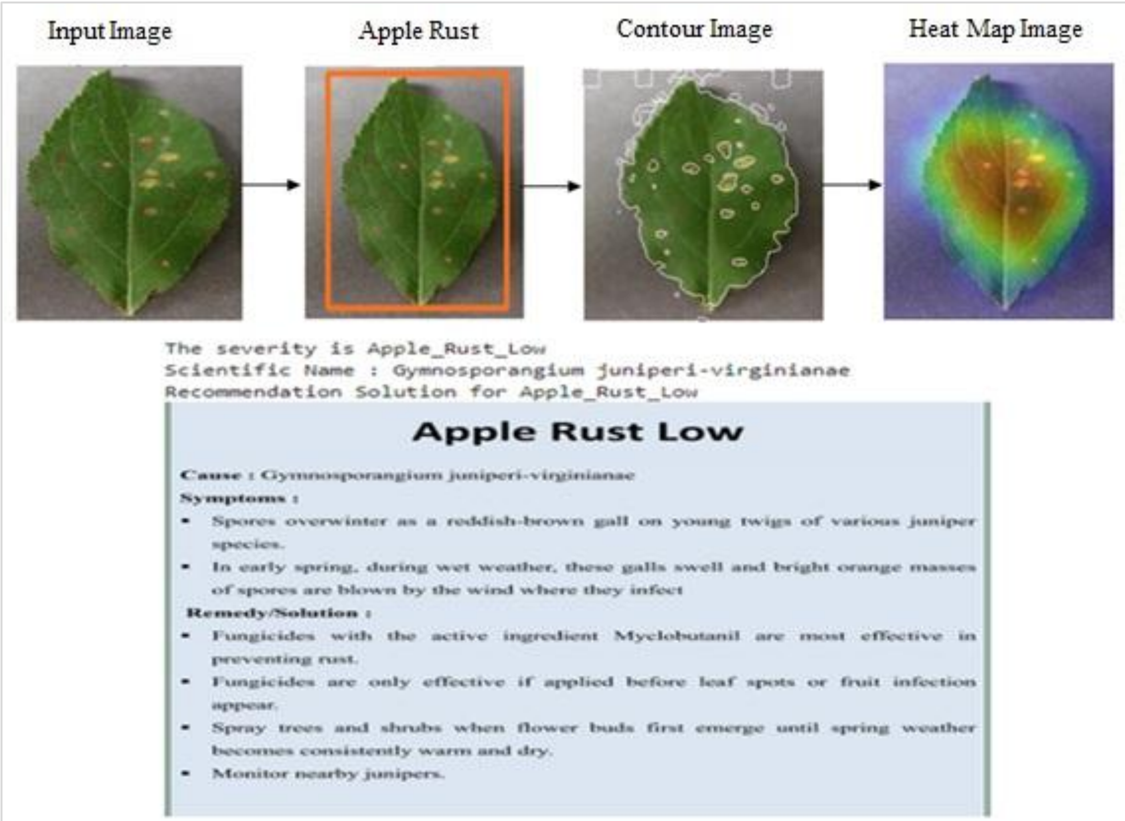


Figure 15 End-to-end leaf disease detection and classification results for apple rust

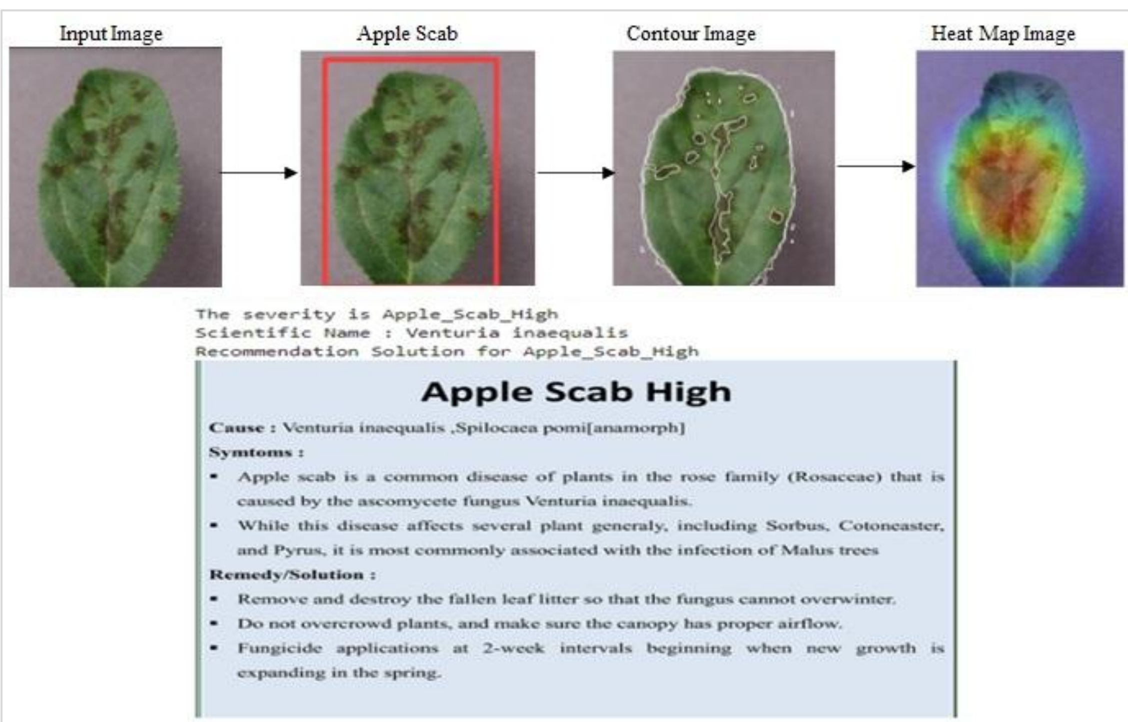


Figure 16 End-to-end leaf disease detection and classification results for apple scab

Table 9 Precision, recall, and F1 score of Multistage Leaf Disease Classification with background

S. No.	Classes	Precision	Recall	F1 score	Support
1	Apple_rust_High	0.854	0.545	0.668	115
2	Apple_rust_Low	0.853	0.648	0.736	140
3	Apple_rust_Moderate	0.556	0.916	0.694	186
4	Apple_Scab_High	0.951	0.583	0.723	115
5	Apple_Scab_Low	0.897	0.931	0.919	202
6	Apple_Scab_Moderate	0.707	0.931	0.813	201
	Accuracy			0.758	959

Table 10 Precision, recall, and F1 score of multistage leaf disease classification without background

S. No.	Classes	Precision	Recall	F1 score	Support
1	Apple_rust_High	0.844	0.958	0.898	207
2	Apple_rust_Low	0.923	0.903	0.913	169
3	Apple_rust_Moderate	0.955	0.903	0.913	194
4	Apple_Scab_High	0.902	0.921	0.914	186
5	Apple_Scab_Low	0.862	0.936	0.898	207
6	Apple_Scab_Moderate	0.988	0.874	0.928	174
	Accuracy			0.910	1137

Table 11 Comparison of F1 Score for multistage leaf disease classification with and without background using ViT

S. No.	Model	Dataset used	F1 score
1	Multistage classification using ViT with background	Apple leaf disease from plantvillage dataset	0.758
2	Multistage classification using ViT without background	Apple leaf disease from plantvillage dataset	0.910

5.Discussion

In this section, a detailed discussion is made based on the results obtained in the above sections. The experimental study performed is multifold. First, disease detection is performed using YOLOv5 deep learning model. Second, background removal is performed using U2-Net in order to correctly classify the diseased leaf. Third, leaf disease stage classification is performed by using the ViT classifier. Fourth, the performance analysis of leaf disease detection, background removal technique, severity classification, and, finally the comparison study is discussed. The disease detection training performance on the PlanDoc dataset is represented through a confusion matrix (see *Figure 7*), where the average mAP values generated for classes apple leaf(leaflet), apple scab leaf(leaflet), and apple rust leaf(leaflet), are 0.258,0.551 and 0.245 respectively at a confidence score of 0.7. Similarly, the performance is shown by the F1 curve and PR-curve (see *Figure 8*) where the maximum F1 score is 0.57 at a confidence score of 0.223 by considering all the classes of PlantDoc dataset and the PR score is 0.623 at 0.5 mAP. The different values of accuracy and loss of leaf disease detection using YOLOv5 are calculated (see *Figure 9*). The training losses are calculated such as box loss is 0.028874, object loss is 0.03059, and classification loss is 0.0066293. Similarly, validation losses are calculated and their values are 0.031752, 0.011021, and 0.041854 respectively. The accuracy value is 0.66002, recall is 0.54152, mAP_0.5 is 0.61814 and mAP_0.5:0.95 is 0.46136. The performance of background removal is performed by using the dice score metric. The Dice score of apple scab is 0.750+-0.0977 and apple rust is 0.756+-0.093 (see *Figure 12*). Finally, the integration results have been shown for apple rust and apple scab in which first the input image is passed to the integration model which will identify as either apple rust or apple scab, and then a contour image is generated which represents the severity of leaf than, the heat map image is generated which represents the color pixel value of hue or intensity and finally the severity of the leaf is shown with a scientific name of the disease and the recommended solution for that particular disease which will be helpful for the farmers to take the action against the disease.

The comparative analysis is performed by considering precision, recall, and F1 score with support value as performance standards. F1 score of different severity levels of apple rust disease with the background are apple_rust_high, apple_rust_low, and apple_rust_moderate is 0.668, 0.736, and 0.694. Similarly, the F1 score of different severity levels of apple scab disease with the background are apple_scab_high, apple_scab_low, and apple_scab_moderate is 0.723, 0.919, and 0.813. So, the overall F1 score of multistage classification of apple leaf with background is 0.738 (see *Table 9*). The F1 score of different severity levels of apple rust disease without background are apple_rust_high, apple_rust_low, and apple_rust_moderate is 0.898, 0.913, and 0.913. Similarly, the F1 score of different severity levels of apple scab disease without background are apple_scab_high, apple_scab_low, and apple_scab_moderate is 0.914, 0.898, and 0.928. So, the overall F1 score of multistage classification of apple leaf without background is 0.910 (see *Table 10*). By comparing both results, the F1 score of multistage classification of apple leaf without background outperforms with a background.

Implications: The implications of this research work are as follows.
Improved accuracy in leaf disease classification: The proposed deep learning approach achieves high accuracy in classifying the severity of leaf diseases in plants. This could lead to more effective and timely management of plant diseases, potentially leading to increased crop yields and reduced economic losses.

Explainability in AI: An explicable AI strategy offers insights into the model's decision-making process. This is important in many applications, such as agriculture and healthcare, where it is critical to know why a model made a certain decision. Transferability of transformer-based models: Modern methods for NLP tasks are used in this article, including the transformer network design. The success of this approach in classifying leaf diseases suggests that transformer-based models can be applied to other image classification tasks, as well. Advancements in plant disease detection: By enabling early identification and management of plant disease, the application of deep learning to plant disease detection has the potential to revolutionize agriculture. This could lead to reduced pesticide use, lower costs, and increased yields.

Overall, leaf disease severity classification with XAI using transformer networks has implications for various fields, including agriculture, AI, and explainability. It demonstrates the potential of deep learning approaches in plant disease detection and management, while also highlighting the importance of XAI in ensuring trust and transparency in AI systems.
Limitation: This work is limited to the severity classification of apple leaf diseases, apple rust, and apple scab. In *Appendix I*, an exhaustive list of acronyms is provided.

6.Conclusion and future work

The primary goal of the proposed work is to find the most cutting-edge deep-learning model for plant leaf disease detection and stage classification based on the severity of the plant leaf. Real-time plant leaf disease detection and stage classification using XAI methodology has been implemented in this research work. This has given an insight into the classification and also provided expert recommendations through the suggested solution to mitigate the leaf disease to increase the end-users' trust and support. The multi-class apple plant leaf diseases are detected and the best classification stage is identified as it belongs to low, moderate, and high severity based on the severity of the infection and a real-time application can be inferred using deep learning techniques. YOLOv5 with various confidence scores was used in the tests using the PlantDoc dataset to detect leaf disease. The maximum detection accuracy for plant leaf disease is 0.443 mAP with a confidence score of 0.7. Background removal is performed using U2-Net architecture for both Apple Scab and Apple Rust diseased leaf. The mean value of the Apple Scab leaf and Apple Rust leaf are 0.750 and 0.756 and the standard deviation (STD) is 0.097 and 0.093 respectively. Gradient-weighted class activation mapping (Grad-CAM), an explainable AI method, was applied to the projected output using a YOLOv5-based validation to foster trust with an F1 score of 91% on both the original and enhanced datasets, the ViT was determined to be the best classification model for diagnosing the infected leaf. The client-server interface is used by the end-to-end model. Through the offered client-server interface, users can upload photos of infected leaves. The end-user received a professional recommendation based on the classification and analysis. The proposed system can make a redolent contribution to agriculture research.

Future work: Even though high accuracy is achieved on the classification dataset, however on the real-time datasets, one can measure recognizable improvements in accuracy. The system can be improved with many different plant species with different leaf diseases, and it is expected to be improved significantly with more and more training data in the future.

Acknowledgment
None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author’s contribution statement

Revanasiddappa Bandi: Worked on the literature survey, methodology, implementation, result analysis, and manuscript preparation. **Suma Swamy:** Worked on the manuscript correction. **Arvind C. S:** Worked on the problem formulation and manuscript correction.

References

[1] https://agricoop.nic.in/Documents/annual-report-2020-21.pdf. Accessed 19 February 2023.

[2] Raza A, Razzaq A, Mehmood SS, Zou X, Zhang X, Lv Y, et al. Impact of climate change on crops adaptation and strategies to tackle its outcome: a review. Plants. 2019; 8(2):1-29.

[3] López MM, Bertolini E, Olmos A, Caruso P, Gorris MT, Llop P, et al. Innovative tools for detection of plant pathogenic viruses and bacteria. International Microbiology. 2003; 6:233-43.

[4] Liang Q, Xiang S, Hu Y, Coppola G, Zhang D, Sun W. PD2SE-Net: computer-assisted plant disease diagnosis and severity estimation network. Computers and Electronics in Agriculture. 2019; 157:518-29.

[5] Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N. PlantDoc: a dataset for visual plant disease detection. In proceedings of the 7th ACM IKDD CoDS and 25th COMAD 2020 (pp. 249-53).

[6] https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset. Accessed 10 February 2023.

[7] Fang Y, Ramasamy RP. Current and prospective methods for plant disease detection. Biosensors. 2015; 5(3):537-61.

[8] Mou L, Meng Z, Yan R, Li G, Xu Y, Zhang L, et al. How transferable are neural networks in NLP applications? Proceedings of the conference on empirical methods in natural language processing 2016 (pp. 479-89). Association for Computational Linguistic.

[9] Dhanya VG, Subeesh A, Kushwaha NL, Vishwakarma DK, Kumar TN, Ritika G, et al. Deep learning based computer vision approaches for smart agricultural applications. Artificial Intelligence in Agriculture. 2022; 6:211-9.

[10] Dhaygude SB, Kumbhar NP. Agricultural plant leaf disease detection using image processing. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2013; 2(1):599-602.

[11] Sethy PK, Barpanda NK, Rath AK. Detection and identification of rice leaf diseases using multiclass SVM and particle swarm optimization technique. International Journal of Innovative Technology and Exploring Engineering. 2019; 8(6S2):2278-3075.

[12] Sebastian VB, Unnikrishnan A, Balakrishnan K. Gray level co-occurrence matrices: generalisation and some new features. International Journal of Computer Science, Engineering and Information Technology. 2012; 2(2):151-7.

[13] Khirade SD, Patil AB. Plant disease detection using image processing. In international conference on computing communication control and automation 2015 (pp. 768-71). IEEE.

[14] Cao D, Chen Z, Gao L. An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks. Human-centric Computing and Information Sciences. 2020; 10(1):1-22.

[15] Guo Y, Zhang J, Yin C, Hu X, Zou Y, Xue Z, et al. Plant disease identification based on deep learning algorithm in smart farming. Discrete Dynamics in Nature and Society. 2020; 2020:1-11.

[16] Cohen R. The chan-vese algorithm. Computer Vision and Pattern Recognition. 2011:1-18.

[17] Hassan SM, Maji AK, Jasiński M, Leonowicz Z, Jasińska E. Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics. 2021; 10(12):1-19.

[18] Rashid J, Khan I, Ali G, Almotiri SH, Alghamdi MA, Masood K. Multi-level deep learning model for potato leaf disease recognition. Electronics. 2021; 10(17):1-27.

[19] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In proceedings of the conference on computer vision and pattern recognition 2016 (pp. 779-88). IEEE.

[20] Redmon J, Farhadi A. Yolo9000: better, faster, stronger. In proceedings of the conference on computer vision and pattern recognition 2017 (pp. 7263-71). IEEE.

[21] Redmon J, Farhadi A. Yolov3: an incremental improvement. Computer Vision and Pattern Recognition. 2018:1-6.

[22] Bochkovskiy A, Wang CY, Liao HY. Yolov4: optimal speed and accuracy of object detection. Computer Vision and Pattern Recognition. 2020:1-17.

[23] https://github.com/ultralytics/yolov5. Accessed 10 February 2023.

[24] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016:1-14.

[25] http://www.pascal-network.org/challenges/VOC/voc2007/index.html. Accessed 15 February 2023.

[26] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/index.html. Accessed 15 February 2023.

[27] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: common objects in context. In computer vision–ECCV, 13th European conference, Zurich, Switzerland, proceedings 2014 (pp. 740-55). Springer International Publishing.

[28] Tan M, Le Q. Efficientnet: rethinking model scaling for convolutional neural networks. In international conference on machine learning 2019 (pp. 6105-14). PMLR.

[29] Malik HS, Dwivedi M, Omkar SN, Javed T, Bakey A, Pala MR, et al. Disease recognition in sugarcane crop using deep learning. In advances in artificial

intelligence and data engineering 2021 (pp. 189-206). Springer Singapore.
[30] Toda Y, Okura F. How convolutional neural networks diagnose plant disease. Plant Phenomics. 2019. 2019:1-14.
[31] Nagasubramanian K, Jones S, Singh AK, Sarkar S, Singh A, Ganapathysubramanian B. Plant disease identification using explainable 3D deep learning on hyperspectral images. Plant Methods. 2019; 15:1-10.
[32] Liu J, Wang X. Early recognition of tomato gray leaf spot disease based on MobileNetv2-Yolov3 model. Plant Methods. 2020; 16:1-6.
[33] Malik A, Vaidya G, Jagota V, Eswaran S, Sirohi A, Batra I, et al. Design and evaluation of a hybrid technique for detecting sunflower leaf disease using deep learning approach. Journal of Food Quality. 2022; 2022:1-12.
[34] Tammina S. Transfer learning using vgg-16 with deep convolutional neural network for classifying images. International Journal of Scientific and Research Publications. 2019; 9(10):143-50.
[35] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: inverted residuals and linear bottlenecks. In proceedings of the conference on computer vision and pattern recognition 2018 (pp. 4510-20). IEEE
[36] Srinidhi VV, Sahay A, Deeba K. Plant pathology disease detection in apple leaves using deep convolutional neural networks: apple leaves disease detection using efficientnet and densenet. In 5th international conference on computing methodologies and communication 2021 (pp. 1119-27). IEEE.
[37] Sibiya M, Sumbwanyambe M. Automatic fuzzy logic-based maize common rust disease severity predictions with thresholding and deep learning. Pathogens. 2021; 10(2):1-17.
[38] Shiferaw D, Hailegnaw A, Assefa A, Abebe D, Dagne E, Fekadie G. LabVIEW based fuzzy logic controller for PLC. In international conference on computer, control, electrical, and electronics engineering 2019 (pp. 1-4). IEEE.
[39] Shi T, Liu Y, Zheng X, Hu K, Huang H, Liu H, Huang H. Recent advances in plant disease severity assessment using convolutional neural networks. Scientific Reports. 2023; 13(1):1-14.
[40] Abd AYM, Caro OJ, Bravo LM, Kaur C, Al AMS, Bala BK. Leaf disease identification and classification using optimized deep learning. Measurement: Sensors. 2023; 25:1-6.
[41] Mohammed L, Yusoff Y. Detection and classification of plant leaf diseases using digital image processing methods: a review. ASEAN Engineering Journal. 2023; 13(1):1-9.
[42] Han K, Wang Y, Chen H, Chen X, Guo J, Liu Z, et al. A survey on vision transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022; 45(1):87-110.
[43] Xu R, Lin H, Lu K, Cao L, Liu Y. A forest fire detection system based on ensemble learning. Forests. 2021; 12(2):1-16.
[44] Bo S, Jing Y. Image clustering using mean shift algorithm. In fourth international conference on computational intelligence and communication networks 2012 (pp. 327-30). IEEE.
[45] Canny J. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1986; 8(6):679-98.
[46] Qin X, Zhang Z, Huang C, Dehghan M, Zaiane OR, Jagersand M. U2-Net: going deeper with nested U-structure for salient object detection. Pattern Recognition. 2020; 106(2020):1-12.
[47] Arvind CS, Prajwal R, Bhat PN, Sreedevi A, Prabhudeva KN. Fish detection and tracking in pisciculture environment using deep instance segmentation. In TENCON region 10 conference 2019 (pp. 778-83). IEEE.

Revanasiddappa Bandi received a B.E. degree in Computer Science & Engineering from PDA College of Engineering, Gulbarga, Karnataka, India, in the year 2008 and M. Tech in Computer Science & Engineering from Dr. Ambedkar Institute of Technology, Bangalore, India in the year 2010, currently pursuing a Ph.D. degree from the Visvesvaraya Technological University, Belagavi, India. He is currently working as Assistant Professor with the Department of Computer Science and Engineering at Dr. Ambedkar Institute of Technology, Bangalore, India. His major research areas are the Internet of Things, Smart Agriculture, Machine Learning, and Artificial Intelligence.
Email: revanasiddappa.cs@drait.edu.in

Dr. Suma Swamy received a B.E. degree in Electronics from Shivaji University, Maharashtra, India, in 1990 and M. Tech in Electronics from Visvesvaraya Technological University, Belagavi, India in year 2006. She has completed Ph.D. degree from Anna University, Chennai, India in year 2014. She joined various different engineering colleges at different positions. She is currently working as Professor with the Department of Computer Science and Engineering at Sir. M Visvesvaraya Institute of Technology, Bangalore, India. Her major research areas are Speech Processing, Data Mining, Big Data, Machine Learning, Natural Language Processing and IoT. She has many national and international research publications.
Email: sumaswamy_cs@sirmvit.edu

Arvind C. S is a Senior Researcher in the Bioinformatic Institute (BII) at the Agency for Science, Technology and Research (A*STAR), Singapore & IEEE Senior Member. He has received his Master of Technology degree in Computer Science Engineering from Visvesvaraya Technological University (VTU), India. He has worked on AI for Mobile Platform, Vision-Based Driver Assistance Autonomous Vehicle Program, Biomedical Imaging, Internet of Things (IoT). He has published over 20 referred articles in IEEE, Elsevier and Springer Journals and several patents. His current research interests include Artificial Intelligence, AI for Radiology, Deep Generative Models, Reinforcement Learning and optimization.
Email: csarvind2000@gmail.com

Appendix I

S. No.	Abbreviation	Description
1	ACO-CNN	Ant Colony Optimization with Convolution Neural Network
2	ADAM	Adaptive Moment Estimation
3	AI	Artificial Intelligence
4	CADS	Computer-Aided Disease Detection System
5	CBN	Cross-Iteration Batch Normalization
6	CNN	Convolution Neural Network
7	COCO	Common Objects in Context
8	CSP	Cross-Stage-Partial Connections
9	DCNN	Deep Convolutional Neural Network
10	Faster R-CNN	Faster Region-Based Convolutional Neural Networks
11	FLOPS	Floating-Point Operations Per Second
12	FPN	Feature Pyramid Network
13	GDP	Gross Domestic Product
14	GLCM	Gray Level Cooccurrence Matrix
15	GPU	Graphics Processing Unit
16	Grad-CAM	Gradient-Weighted Class Activation Mapping
17	HSV	Hue Saturation Value
18	ILSVRC	ImageNet Large Scale Visual Recognition Challenge
19	IOU	Intersection Over Union
20	IQR	Inter Quartile Range
21	mAP	Mean Average Precision
22	Multiclass-SVM	Multiclass Support Vector Machine
23	MLP	Multi-Layer Perceptron Layer
24	MS	Microsoft
25	NLP	Natural Language Processing
26	PAN	Pan Aggregation Network
27	PLD	Potato Leaf Diseases
28	PR	Precision-Recall
29	PSO	Particle Swarm Optimization
30	ReLU	Rectified Linear Unit
31	RGB	Red Green Blue
32	ROC	Receiver Operating Characteristic
33	ROI	Region of Interest
34	RPN	Region Proposal Network
35	RSU	Residual U-Block
36	SGD	Stochastic Gradient Descent
37	SSD	Single Shot Detector
38	STD	Standard Deviation
39	SVM	Support Vector Machine
40	VGG	Visual Geometry Group
41	ViT	Vision Transformer
42	VOC	Visual Object Classes Challenge
43	WRC	Weighted-Residual-Connections
44	XAI	Explainable Artificial Intelligence
45	YOLO	You Only Look Once