agriculture

MDPI

Article

Real-Time Plant Health Detection Using Deep Convolutional Neural Networks

Mahnoor Khalid [1], Muhammad Shahzad Sarfraz [1], Uzair Iqbal [2], Muhammad Umar Aftab [1], Gniewko Niedbała [3,*] and Hafiz Tayyab Rauf [4,*]

[1] Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Chiniot-Faisalabad Campus, Chiniot 35400, Pakistan
[2] Department of Artificial Intelligence and Data Science, National University of Computer and Emerging Sciences (NUCES), Islamabad 35400, Pakistan
[3] Department of Biosystems Engineering, Faculty of Environmental and Mechanical Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland
[4] Independent Researcher, Bradford BD8 0HS, UK
* Correspondence: gniewko.niedbala@up.poznan.pl (G.N.); hafiztayyabrauf093@gmail.com (H.T.R.)

Abstract: In the twenty-first century, machine learning is a significant part of daily life for everyone. Today, it is adopted in many different applications, such as object recognition, object classification, and medical purposes. This research aimed to use deep convolutional neural networks for the real-time detection of diseases in plant leaves. Typically, farmers are unaware of diseases on plant leaves and adopt manual disease detection methods. Their production often decreases as the virus spreads. However, due to a lack of essential infrastructure, quick identification needs to be improved in many regions of the world. It is now feasible to diagnose diseases using mobile devices as a result of the increase in mobile phone usage globally and recent advancements in computer vision due to deep learning. To conduct this research, firstly, a dataset was created that contained images of money plant leaves that had been split into two primary categories, specifically (i) healthy and (ii) unhealthy. This research collected thousands of images in a controlled environment and used a public dataset with exact dimensions. The next step was to train a deep model to identify healthy and unhealthy leaves. Our trained YOLOv5 model was applied to determine the spots on the exclusive and public datasets. This research quickly and accurately identified even a small patch of disease with the help of YOLOv5. It captured the entire image in one shot and forecasted adjacent boxes and class certainty. A random dataset image served as the model's input via a cell phone. This research is beneficial for farmers since it allows them to recognize diseased leaves as soon as they noted and take the necessary precautions to halt the disease's spread. This research aimed to provide the best hyper-parameters for classifying and detecting the healthy and unhealthy parts of leaves in exclusive and public datasets. Our trained YOLOv5 model achieves 93 % accuracy on a test set.

Keywords: plant health detection; precision agriculture; deep learning; object detection; YOLOv5

Citation: Khalid, M.; Sarfraz, M.S.; Iqbal, U.; Aftab, M.U.; Niedbała, G.; Rauf, H.T. Real-Time Plant Health Detection Using Deep Convolutional Neural Networks. *Agriculture* **2023**, *13*, 510. https://doi.org/10.3390/agriculture13020510

Academic Editor: Ritaban Dutta

Received: 18 January 2023
Revised: 16 February 2023
Accepted: 17 February 2023
Published: 20 February 2023
Corrected: 27 December 2024

1. Introduction

It is challenging to recognize plant diseases by optically analyzing their signs on plant leaves. Skilled agronomists and plant pathologists frequently require help to accurately diagnose certain diseases due to the diverse array of cultivated plants and phyto-pathological issues, resulting in incorrect diagnoses and treatments. Entomologists who are requested to make these diagnoses by visual examination of diseased plant leaves would greatly benefit from the development of an ASO (automated systems operation) to identify and diagnose plant diseases [1]. Humans eat food that comes from plants. Furthermore, because plants create oxygen, they aid in maintaining the oxygen in the air.

Without agriculture, the life we live would not be possible. All the goods we use daily, such as oil, firewood, fiber, pesticides, medicine, and rubber, are extracted from plants.

Plants, crops (fruits, vegetables, etc.), and the natural world are significant to humans. Engaging with nature is crucial for improving an individual's quality of life and delivering various measurable advantages to human beings, including psychological/cognitive advantages [2]. A plant comprises several parts, such as leaves, flowers, stems, and roots. A farmer may cultivate many plants, but diseases can impede their growth. Disease attack is one of the primary reasons that lead to plant loss. Each year 10–16% of plant production is reduced due to disease [3]. In past decades, the health consequences of exposure to nature have been described in detail. However, the role of plants, such as money plants, has received enormously little interest, as compared to the range of crop studies. Urban people spend 80–90% of their lives in houses, offices, schools, etc. Good environments are very important for their health. Indoor plants play an essential role in a good and healthy environment, but their impact on the surroundings and human beings has not yet been quantified [2]. Plants are crucial for removing harmful emissions from the atmosphere and enhancing the ecosystem as well as providing a positive psychological effect, increased health, and a comfortable indoor environment. The above studies have shown that plants benefit humans, so caring for plants is also essential. However, there needs to be more research conducted regarding the money plant.

Currently, several strategies for minimizing plant disease include the removal of damaged plant leaves, mechanical cultivation, and the use of various pesticides. Using the services of an agricultural professional is a simple way to detect plant disease. However, manual disease detection takes a long time and is arduous work. The typical strategy is to use pesticides [4,5]; however, excessive pesticide use may enhance plant growth while harming plant quality. However, spraying more pesticides on plants before even assessing the amount of pesticide required for a specific crop could negatively affect the environment and human health [6].

However, plant disease recognition is more accessible through machine learning. The use of this technique has been identified as a vital advancement and management success for plant disease. The agriculture sector's productivity has grown as a result as well. Additionally, image processing methods have been added to this technology, which has advanced during the last three years to its present state [7,8]. The nation's problems, such as lurgies affecting plants and humans, could be mitigated. Once the unhealthy plants were recognized, they covered a large region.

Machine learning (ML) has been widely employed in the world today. AI, known as ML, enables machines to interact with people and understand their needs. Additionally, it enables machines to perform actions usually performed by people. Several issues impact the reliability and performance of this technology, making it challenging for ML methods to identify specific disorders. Figure 1 shows the traditional method of image processing.

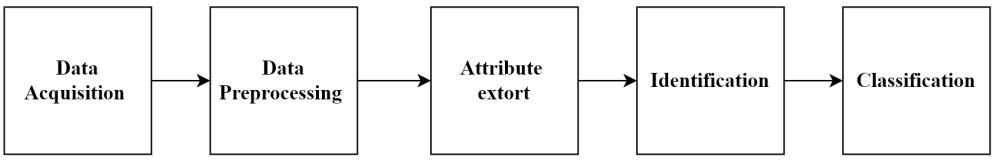

Figure 1. Traditional image pre-processing techniques: The basic method of identifying plant diseases using conventional image recognition processing technologies.

The first problem was the computational time involved with machine learning and deep learning because some methods used to diagnose such diseases must be updated as they rely on obsolete information. Another problem has been segmentation reactivity [9], which refers to the high sensitivity and precision in the relevant field that is required (ROI). A significant amount of resources are required to create and implement the bulk of machine learning and deep learning tasks. Organizations that use this technology for people and plants are frequently supported by non-government organizations, which may affect the development and use of this technology.

To identify diseased leaves, image recognition may be performed. According to background research, by scanning images of infected and healthy leaves, experts in this field have been able to compare them accordingly [10]. Several traditional image processing techniques were used. The image processing had the following steps, e.g., the images were segmented first, then the plant disease features were retrieved, and finally, the disease was categorized. This research developed an attribute image-based method for classifying wheat plant diseases and used an SVM to diagnose the condition [11] successfully. The capability for generalizing new datasets has to be enhanced since attribute information can only be learned superficially. Deep-learning methods, however, are being used in farming research more often, as they can rapidly retrieve deep feature data and are quicker and more accurate than the standard machine learning (ML) algorithms [12,13]. Researchers have created a rider neural network based on the sine-cosine algorithm and discovered that the classifier's identification performance increased significantly [14]. As has been demonstrated, deep-learning models have proliferated over the last decade. Experts have been assisted by the numerous methods used in machine learning (ML) and deep learning (DL) to quickly identify the causes of plant diseases and evaluate their symptoms. In summary, deep learning has shown successful outcomes in the identification of plant diseases. Our research was conducted on categories of plants because both plants and crops have the same importance for the environment and humans. For the sake of world health and well-being, it is essential to identify plant diseases accurately.

Figure 2 below depicts three of the most prevalent diseases that affect plants. Each has distinctive signs and side effects on the leaf; these can be used to differentiate and label the infections visually by the human eye and automatically by algorithms. In today's ever-changing environment, early detection of disease and early prevention is critical to avoid issues that could otherwise arise. Figure 2 shows two types of data depicting the classes in the dataset: the first depicts an unhealthy sample while the second shows healthy sample.Therefore, the main goal was to achieve accurate detection between diseased and healthy, and a deep-learning network-based YOLOv5 model was used in this study. The described model has been useful in plant and crop plantations and agricultural production, according to experiments conducted on images with complicated backdrops. Hence, deep learning (DL) is the most accurate and precise way to identify diseases in plants with the best results. The following is a summary of this work's significant contributions:

- A dataset with several scenarios and sizes was created. There were a variety of sophisticated backdrops, with varying lighting and perspectives, that featured images of damaged leaves. This offered the optimal information to make plant disease research easier.
- This study provided thorough, detailed literature on the methods currently used in plant disease identification. It also reviewed the literature on the datasets used in the research and provided a comparative analysis that identified various studies' advantages and disadvantages.
- This study used different-deep-learning algorithms for the classification of plant anomalies.
- This study established the hyper-parameters for the applied deep algorithm for comparison with state-of-the-art plant disease classification algorithms (standard machine learning (ML) approaches)
- This study evaluated the applied deep-learning algorithm with standard efficiency parameters.
- This research evaluated 4 different target detection techniques, and the results of the trials demonstrated that the suggested approach achieved an mAP of 93.1% on both exclusive and public datasets at 120 frames per second (FPS). Precision planting, visual management, and intelligent decision-making were all features of the YOLOv5 algorithm for economic productivity.

The following represents the paper's essential topics. Before presenting the study objectives, the relevant material and related work are introduced. Next, the model concept and improvement, model training and testing, research object (dataset), and operation

procedure are introduced. Then, the findings are examined to demonstrate the viability and progress of the model used in this study. Finally, the overall research is reviewed, and suggestions for further research are offered.

Figure 2. Data set visual representation.

2. Related Work

This research reviewed the detection of healthy and unhealthy leaves in plants; the different databases used to collect datasets, feature extraction, and feature selection techniques; and the machine-learning and deep-learning models reported in the literature.Furthermore, this research included an overall pipeline strategy employed in previous studies. Some researchers worked on different stress responses in plants using deep-learning models. Plants must handle stress due to various types of environmental factors, such as water stress, dust, and diseases caused by bacteria that affect their health. In the past few years, machine-learning techniques have been used for solving such issues, but deep learning, CNN, and other algorithms have also been used for detection, evaluation, and comparison [15].

Meanwhile, other research has examined water stress in plants using deep-learning techniques. Plant growth is controlled directly by plant water stress and only indirectly by soil water stress, and this has resulted in losses of up to 90% due to water stress and heat stress. Image processing is a directed way to measure the water stress in plants beyond the limitations of traditional image processing. This researcher used deep-learning techniques (Alex Net, Google Net, and Inception V3) on maize (*Zea mays*), okra (*Abelmoschus esculentus*), and soybean (*Glycine max*) crops to identify water stress based on a 1200 images dataset [16].

Maize is the most cultivated crop in the world [17]. Its stable production has a significant influence on food security. In addition, maize is sensitive to drought stress. Many studies have shown that water stress during the agronomy and tasselling stages reduced plant yield and caused a 29–32% final dry. Drought has become a vital factor that lemmatizes maize yield. This research mainly focused on maize drought detection. Many previous studies have explained the traditional detection methods based on power, low-cost, and manual experiments.In recent years, image processing techniques and computer vision technology have become widely used. Image processing is low cost, low power, and convenient for real-time analysis techniques. According to the research, the water supply in the two weeks before and after the pollination period determines the final yield. In this study, they used different directions and wavelengths of Gabor filters. The results of the experiments were 98.84% [18].

Avocado is a tropical fruit with a significant economic value in Florida [19]. The research presented and evaluated an automated detection technique for avocado trees. Remote sensing techniques have been used for detection and evaluation in order to compare

healthy and unhealthy trees. In this study, laurel wilt (LW) disease was the focus, and they differentiated sick and healthy trees (H). The detection of LW during its early stages is challenging because it shows symptoms similar to other stress factors: nutrient deficiency and salt damage. The accuracy of the experiment was 99%. Therefore, low-cost remote techniques can be utilized to differentiate healthy and unhealthy plants.

However, this study focused mainly on plant disease automation in farming. It is a major concern in many countries, as food consumption is growing at a rapid rate due to population growth. Furthermore, modern technologies have improved the efficiency and accuracy of disease detection in plants and animals. The detection procedure is the first step in a workflow designed to combat diseases and limit their spread. The research focused mainly on details about diseases and the implementation of artificial intelligence to detect them quickly. Moreover, machine learning and deep-learning models are used for the automatic detection of plant disease. Various datasets have also been examined for achieving better outcomes for the research community [20].

Furthermore, in other research [21], PlantDoc showed a crop production loss of 35% as a result of plant disease. The early diagnosis of plant infection is still challenging due to a need for more tools and knowledge. This research examined the possibility of using computer vision technologies for low-cost early diagnosis of plant diseases. The main contribution of this research was the PlantDoc dataset that was mentioned in this research.

Previous research [22] has shown the importance of AI in different fields, such as in medical communication, object recognition and detection, etc. This research was only focused on bell pepper.Usually, bell pepper farmers are unaware if their plants are affected by bacterial spot disease. The solution is early detection of infectious spot disease in bell pepper plants. Bacterial spot disease in a bell pepper was detected using YOLOv5-based symptoms on the leaves. They could detect even a small patch of disease faster and more precisely using YOLOv5, and it enabled them to detect diseases during the early stages of development and take appropriate steps to avoid disease spread.This research developed a technique for identifying bacterial spots in bell pepper plants using farm images.

YOLO single-stage real-time object detection has demonstrated the importance of the YOLO principle of object detection [23]. A single-stage network that forecasts the class probability for multiple boxes is known as YOLO. The YOLO network captures the whole image during training. In the aforementioned paper, they discussed the benefits and difficulties of the YOLO algorithm. They contrasted the usual deep-learning methods with YOLO. They noted that YOLO was efficient because it approached object identification as a straightforward regression issue during comparisons. A simple network was optional. They highlighted the core network's 45 FPS (frames per second) speed. More rapid models have been capable of exceeding 150 frames per second. The mean average accuracy was twice as high as other widely used identification techniques. Background inconsistencies were significantly lower, as compared to other deep conventional algorithms, such as faster R-CNN (regions with CNN). There were some drawbacks, as well. Although their method recognized images rapidly, it lacked sufficient accuracy.

In the article YOLOv2: Lighter, quicker, stronger [24], Joseph Redmon proposed a new model that could fix the flaws in the previous version. YOLOv2 aided in resolving the previous version's drawbacks, with its relatively low recall and error analysis in localization. This improved YOLOv2. In YOLOv2, Darknet-19 was implemented. Joint classification and identification, together with hierarchical classification, improved YOLOv2.

A novel YOLO model (YOLOv3) was proposed that advanced the success of YOLOv2 [25]. YOLOv3 had three times the accuracy of conventional approaches, as compared to traditional networks. YOLOv3 yielded good results, as compared to YOLOv2; however, YOLOv3 had limitations, and the component did not operate as intended for linear x y prediction, IOU threshold, and ground-truth assignment. The authors of [26] presented an implementation of the latest iteration, YOLOv4, in YOLOv4: Maximum speed and efficiency of object detection. YOLOv4 increased the accuracy of object recognition. In addition, it raised the FPS and YOLOv3 mean average accuracy by 10% and 12%, respectively.

In another study of deep models that was specifically focused on crops and also relevant to agriculture, the researchers collected data exclusively before deploying several deep and machine learning models to achieve state-of-the-art results [27,28].

3. Comparative Analysis of Selected Study

This study included a comprehensive literature review of previous object detection experiments, as shown in Table 1. The main challenge was to observe all existing models and then compare them with our YOLOv5 model results. After reviewing the existing studies shown in Table 1, we filtered the studies based on our objectives and performed a comparative analysis. A comparative analysis of these studies is shown in Table 2. The different columns show the reference number of the study, the problem under consideration, and the model(s) used to resolve the problem or proposed as a solution. Table 2 shows the analysis and comparisons of the existing problems and the models proposed as solutions.

Table 1. Summary of related work on the identification of plant diseases.

References	Year	Methodology	Dataset Size	Accuracy
[1]	2018	Deep learning	87,848 images	99.53%
[29]	2021	CNN	20,636	98.029%
[30]	2018	Google net Reset	54,306	99.35%
[31]	2018	ANN	Kaggle dataset	80%
[32]	2007	IOT	Custom data	Good Acc.
[33]	2018	3D leaf tracking	12 plants	comparison
[34]	2019	HSI	6 plants	comparison
[35]	2019	Remote sensing	Sentinel-2	Fast, accurate
[36]	2019	Satellite images	Landsat 8	Fast, accurate
[37]	2016	Machine learning	CR262, MTU	comparison
[38]	2019	Alex net	Apples, cherry	Layers convolution
[39]	2018	Alex Net	Tomato leaves	—
[40]	2019	SSD	Banana	RNN Improve

Table 2. Comparative Analysis of Selected Studies.

Ref	Model	Problem	Dataset
[15]	CNN	Plants emotions detection	Kaggle dataset
[16]	Google Net Alex Net Inception V3	Disease detection using deep learning	Custom dataset
[19]	Image processing	Disease Detection in Avocado	Custom dataset
[21]	Mobile Net Faster RCNN	Early plant disease detection	Custom dataset
[22]	YOLOv5	Bell-pepper disease detection	Bell-pepper custom dataset
[25]	YOLOv3	Object detection in images	Kaggle dataset
[25]	YOLOv4	Object detection in images	Kaggle dataset

4. Materials and Methods

The methodology explains and discusses the proposed solution, data collection, pre-processing, model choice, training, and evaluation. The proposed work was based on deep-learning approaches and discussed this approach in detail. A comparative analysis

compared the proposed approach and traditional deep-learning methods. In the section on pre-processing, novel techniques were used for better results. Furthermore, because CNN has been the most frequently used model in image classification, it also assisted in producing accurate results. The proposed models were evaluated by applying techniques and comparing the results. Figure 3 shows a step-by-step procedure that was used in the plant disease detection and classification process. Furthermore, after gathering the data, they were split into two parts, 80/20 training and testing, respectively. Deep-learning (DL) models were then trained, either from scratch or using a learning strategy, and their training plots were obtained to assess the model's relevance. The next phase involved classifying the images using performance matrices, and the last step involved localizing the images using visualization techniques. Several phases were involved in identifying unhealthy plant leaf regions, which are shown in Figure 3.

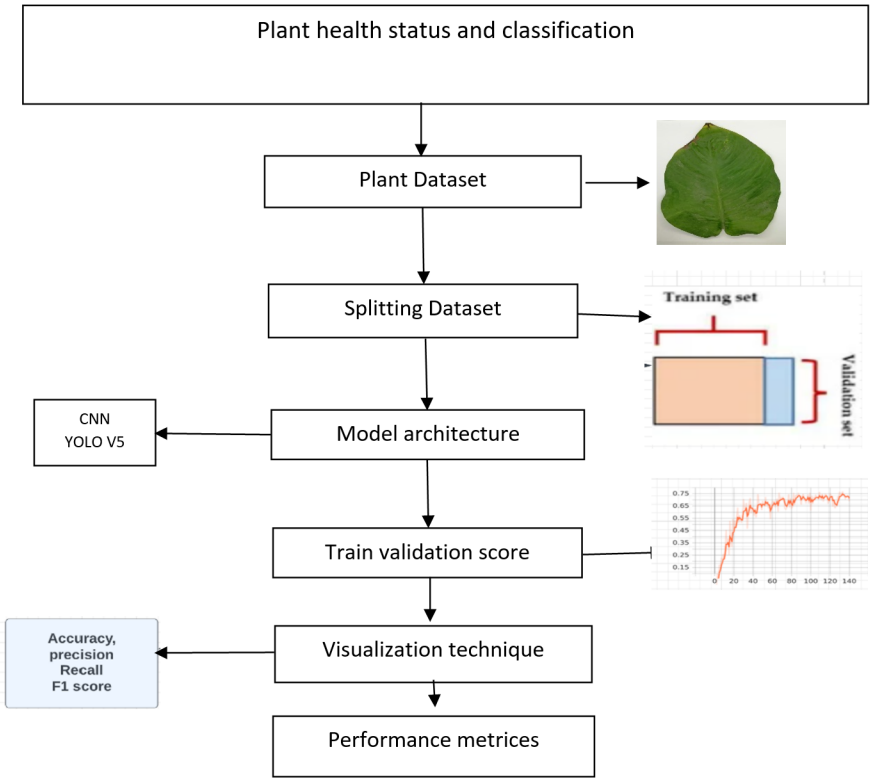

Figure 3. Plant disease identification step-by-step process.

4.1. Experimental Material

The images of the money plant leaves were collected from the University of Agriculture in Faisalabad City, Pakistan, and also from FAST National University of Computer and Emerging Sciences, Faisalabad Chiniot campus, Pakistan. The University of Agriculture is at 31.4300 N, 73.0859 S, and the Fast Chiniot campus is at 31.6076 N, 73.0751 S, with an annual temperature of 15/27 °C. Money plant diseases are typically caused by high humidity and warm temperatures. The dataset was collected in a controlled environment. A HUAWEI/DUAL Lens was used for photography, with an image resolution of 1080 × 2340 pixels and a 19.5:9 aspect ratio. The acquisition of the dataset was one of the key challenges we experienced while working on this research. Using a mobile camera (HUAWEI/DUAL Lens) consumed a great deal of time while capturing images for our exclusive dataset. All the above work was a difficult task for us. Therefore, this research also used the additional dataset offered on the Kaggle website. This research used a public dataset that was collected by Kaggle. The next step was to label the data after obtaining the dataset.

4.2. Preprocessing on Exclusive Data Set

In this research step, the labelIMG tool, as shown in Figure 4, labeled the images. For tool installation, we used Python's pip install labelIMG instruction to install labelIMG. After the installation of labelIMG, the process began with labeling the images. We made bounding boxes around regions of interest (ROI) in the labeling process.

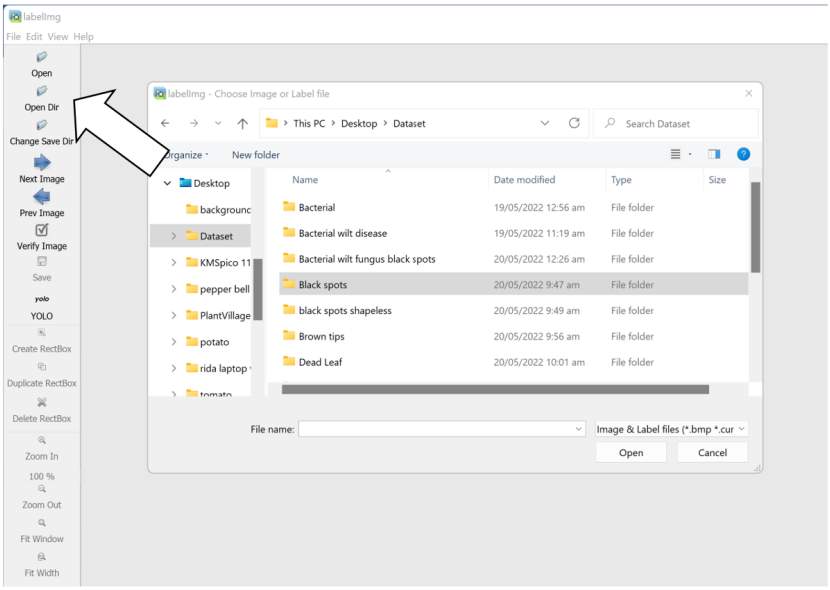

Figure 4. Image selection for labeling.

Briefly, bounding boxes are placed around the unhealthy parts, as shown in Figure 5. We then received a text file as an output after correctly labeling images. Table 3 represents the classes used in our research.

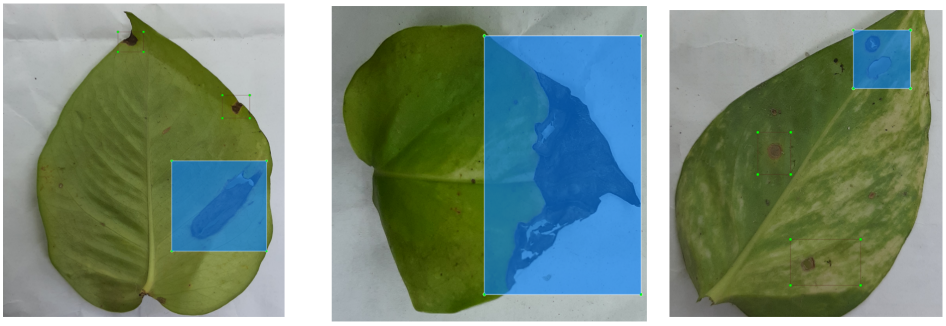

Figure 5. Selected image with unhealthy tag.

Table 3. Dataset Classes.

Class No	Class	Class No	Class
0	Un-Healthy	1	Healthy

One image had multiple bounding boxes depending on the leaves' health condition. The text file had two classes, healthy and unhealthy, with 0 and 1 decimal values. The bounding box class was represented by the first decimal value followed by the centers of the x and y axes, and then the dimensions. The X and Y axes cross at the center point for bounding boxes.These values have been standardized between 0 and 1. We divided the values by the width and height of the image to achieve this. This was conducted rather than using a random integer since it was simpler to estimate values for the network between 0

and 1. The number of bounding boxes drawn determined the number of lines there were in the text file. The dataset was divided into training, testing, and validation. Table 4 shows the total number of training, testing, and validation images.

Table 4. Total Dataset.

Training Sample	Testing Sample	Validation Sample	Total Sample
2000	206	105	2311

After labeling the dataset as healthy or unhealthy, all the above work was conducted with the help of an anaconda prompt. All the above tasks performed in labeling were conducted with labelIMG.

4.3. Preprocessing on Public Data Set

This research work also used a public dataset from the Kaggle site, as shown in Figure 6. The plant village dataset had 28 different classes of plants with 54,309 different images. This research was focused on bell pepper and potato leaves. A total of 1000 images were collected. The images were preprocessed to normalize their dimensions, reduce noise, remove backgrounds, and minimize unwanted distortions. Each dataset image was annotated with the labelIMG tool, and many annotation tools were used, such as coco json, TensorFlow object detector, scale, label box, etc. Therefore, labelIMG tool assisted in making bounding boxes around the leaves in all images. In real life, the images could include many leaves or a mix of infected and healthy leaves. All of the leaves in the images were explicitly labeled with their relevant healthy or unhealthy classes. The complete leave was present in the box while labeling the boxes, and the bounding box area was at least 1/8 (roughly) of the image size. After labeling, all the coordinates of boxes in an image and their related class labels were saved individually in a YOLO file for each image. The resulting pictures were utilized in the next phase as input.

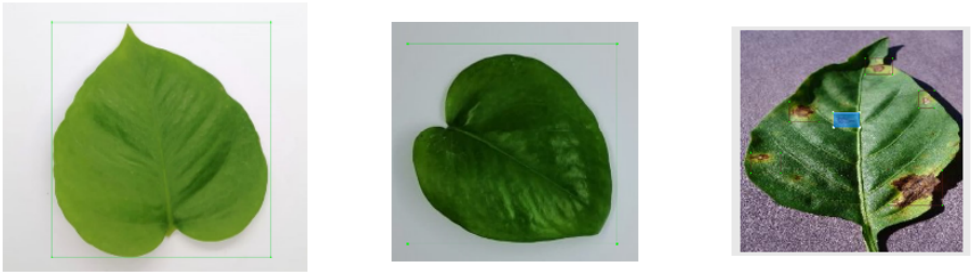

Figure 6. Plant village dataset with bounding boxes.

4.4. Plant Disease Detection and Classification Model Implementations

4.4.1. EfficentDet

Google researchers recently unveiled Efficient Net, which is a convolutional network. The three sections are part of the object detector group known as EfficentDET. Efficient NET serves as the foundation of Google's information collection process. However, it also recycles the milestones from ImageNet's pre-trained network because it employs the same spacing scaling parameters as EfficientNet-B0 through B6. The bi-directional feature network serves as the feature chain for EfficentDET, as shown in Figure 7. BIFPN (bi-directional feature pyramid network) is used for quick and simple 2D feature engineering. The bi-directional feature pyramid uses scales ranging from levels 3 to 7, and the merging process is performed several times. The magnitude levels identify objects in the image that are of various sizes and densities. Image quality and size are directly related to the scaling factor. Consequently, p3 > p4 > p5 > p6 > p7 and p3 > p4 > p5 > p6 > p7 are used to indicate the resolution and quality of objects that were discovered. The bi-directional feature

network is shown in Figure 7. The data flow direction of this method was bi-directional, as shown by the top-down and bottom-up approaches.

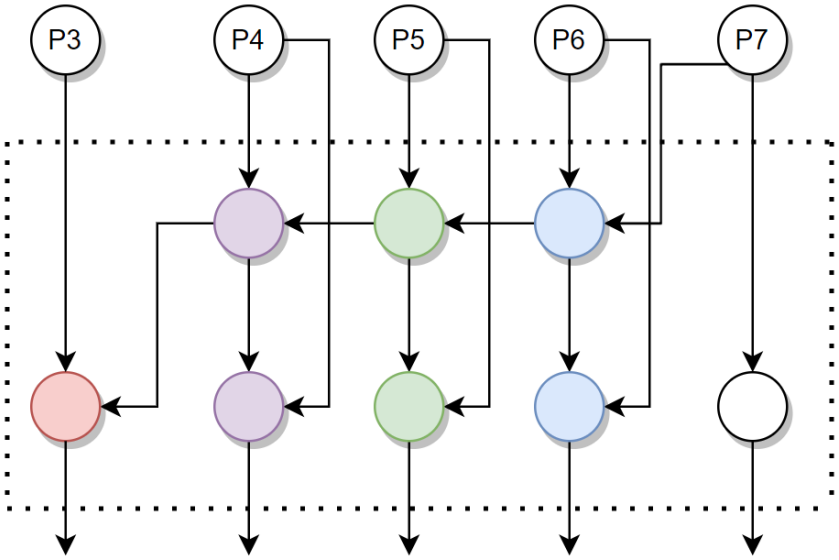

Figure 7. EfficentDet model architecture.

4.4.2. FasterRCNN

Faster R-CNN, the most popular modern variant of the R-CNN series, was released for the first time in 2015. The object proposal method was the only independent network component in the fast regional convolutional network. Faster regional-based CNN and simple RCNN both employ object detection algorithms that are dependent on the hardware capacity. The CPU-based selective search technique, which processes one picture in approximately two seconds, is shown in Figure 8. Moreover, it employs a region proposal network (RPN) to provide object detection recommendations. This improves feature representation overall by reducing the object proposal time per picture from 2 s to 10 milliseconds and enabling the object detection step to share layers with the succeeding detection stages.

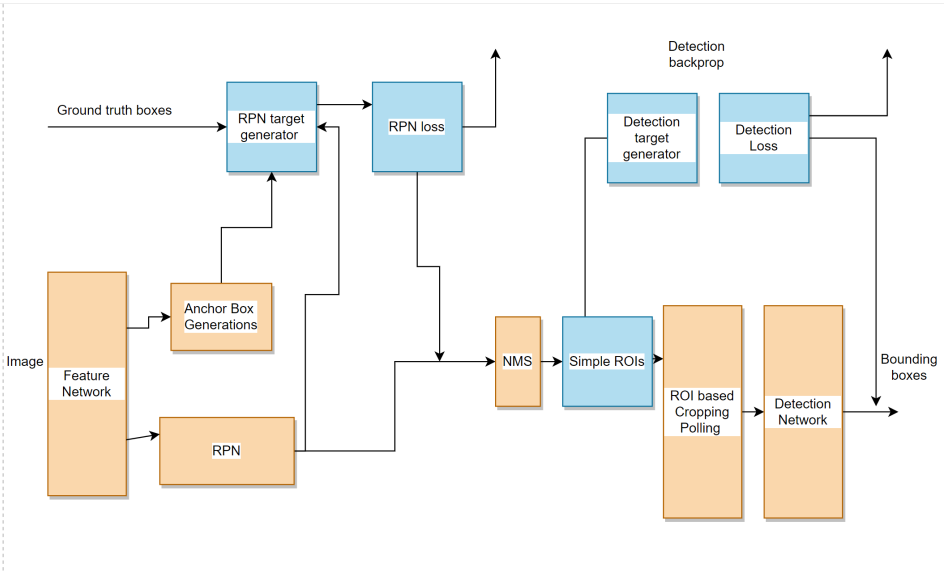

Figure 8. The architecture of Faster R-CNN.

4.4.3. The Principal of YOLOv5 Model

The two-stage object detection methodology led the market before YOLO [25]. It located regions using region of interest-based classifiers and passed those areas on to a stronger classifier. This approach used a lot of resources and necessitates several runs, yet it produced reliable results with high mAP values. The picture is first separated into columns, each of which has an identical SxS-sized dimensional area.

$$S \times S \tag{1}$$

The next step is for each cell to identify and pinpoint the items it contains using the bounding box dimensions, the object name, and the likelihood that the object is present in the columns. Each column "works by itself", processing the grid concurrently while using fewer computer resources as well as training and inference time. In addition, YOLO outperforms other real-time object identification algorithms and produces state-of-the-art results. Furthermore, the output dimensions are

$$S \times S(B \times 5 + C) \tag{2}$$

YOLO Versions

There are six different models introduced to date in the YOLO series (e.g., YOLOv1, YOLOv2, YOLOv3, YOLOv4, YOLOv5, YOLOv6). Our research focused on the YOLOv5 and below series and had better results than the traditional machine-learning model.

YOLOv5 Module

Glenn Jocher presented the one-stage target identification method known as YOLOv5 [31] in 2020. Four network model variations of the YOLOv5 model were distinguished based on differences in network depth and height: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5s network had the highest computational speed but the lowest average processing, whereas the YOLOv5x network exhibited the opposite traits. The YOLOv5 network's model size was around one-tenth that of the YOLOv4 network. Its detection and localization abilities were faster, and its precision was on par with YOLOv4.

4.4.4. YOLOv5 Architecture

The YOLOv5 model in Figure 9 included three crucial components, similar to other single-stage object detectors.

1. Backbone
2. Neck
3. Head

The essential purpose of the model backbone is to extract significant characteristics from an input image. The CSP network was deployed as the backbone to extract important attributes from an input image in YOLOv5. CSPNet demonstrated a considerable reduction in processing time. The primary purpose of the model neck is to produce feature pyramids. Feature pyramids assist the model in making suitable object scaling generalizations. It facilitates recognition of the same item at various sizes and scales. Models perform effectively on unobserved data due to the usage of feature pyramids. Other models such as FPN, BIFPN, and PANet use other feature pyramid methodologies. PANet was utilized in YOLOv5 as a neck to obtain feature pyramids. The last step was the final detection step, which was carried out using the model head. The final output vectors were generated with class probabilities, object scores, and bounding boxes after anchor boxes were applied to the feature.

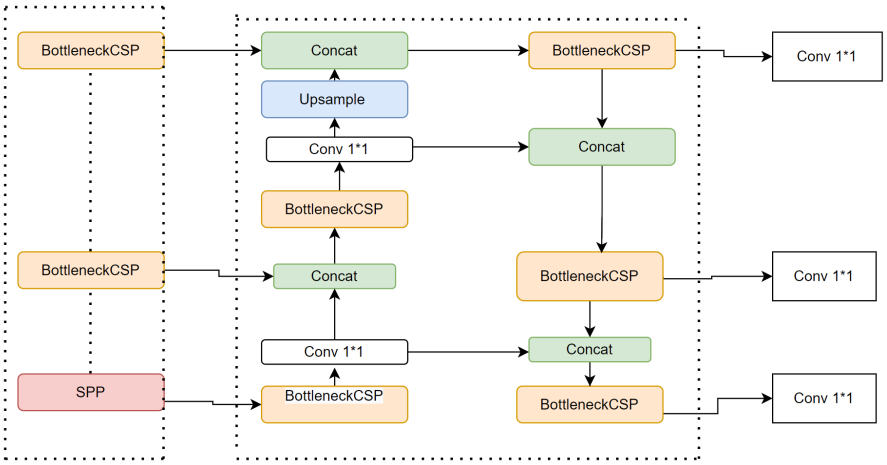

Figure 9. Overview of YOLOv5.

4.4.5. Activation Function

In every deep neural network, the selection of the activation function is of great concern. Many activation functions, such as Leaky ReLU, Mish, and Swish, have recently been developed.

4.4.6. Optimization Function

Optimization was conducted with the assistance of SGD and ADAM.

4.4.7. Cost Function or Loss Function

A compound loss was produced for the YOLO algorithms based on the object score, class probability score, and bounding box regression score. For the loss computation of class probability and object score, ultralytics adopted the binary cross entropy with the logit function from PyTorch. As compared to earlier versions of the YOLO series, version 5 provided superior detection accuracy, a lightweight design, and a quick detection time. Accuracy and effectiveness were important for identifying plant diseases. Therefore, the YOLOv5 model improved disease detection in the money, potato, and bell pepper plants. The model architecture is shown in Figure 9.

4.5. Training Exclusive and Public Datasets on YOLOv5

The following stages were involved in training a custom YOLOv5 model:

1. First step was the environment configuration for YOLO.
2. Obtaining the YOLOv5 repository and installing plugins were the initial steps. As a result, the programming framework was prepared for the execution of instructions for object identification training and inference.
3. We trained a model on the free training environment provided by Google Collab.
4. Google Collab was likely operating on a Tesla P100 GPU.
5. Next, we downloaded the custom data from Roboflow in a YOLOv5 format.
6. After labeling the data, it was then exported to Roboflow. After uploading data into Roboflow, it was converted into one of these formats (VOC XML, coco json, TensorFlow object detection, etc.).
7. After uploading the data, we selected the preprocessing steps and augmentation.
8. Roboflow automatically divided the data into training, testing, and validation sets.
9. After annotating the images, we chose the YOLOv5 pyTorch format.
10. After the format steps, Roboflow provided a key or PIP package, as shown in Figure 10.

```
#from roboflow import Roboflow
#rf = Roboflow(api_key="YOUR API KEY HERE")
#project = rf.workspace().project("YOUR PROJECT")
#dataset = project.version("YOUR VERSION").download("yolov5")
```

Figure 10. Roboflow keys.

The above YOLOv5.YAML file name data.yaml file contained information about the exclusive dataset, as well as the location of the YOLOv5 images. With the data.yaml file, the training process could start immediately:
Img: Image size as the input image
Batch size: determine the length of the batch for training
Epochs: define the training steps
Cfg: model configuration

During training, mAP @ 0.5 minimum average precision was the major concern to determine the detector's performance.

YOLOv5 Evaluation and Validation Metrics

Figure 11 shows verification metrics to determine the training process performance. Once the training process was completed, validation accuracy was performed, as shown in Figure 11.

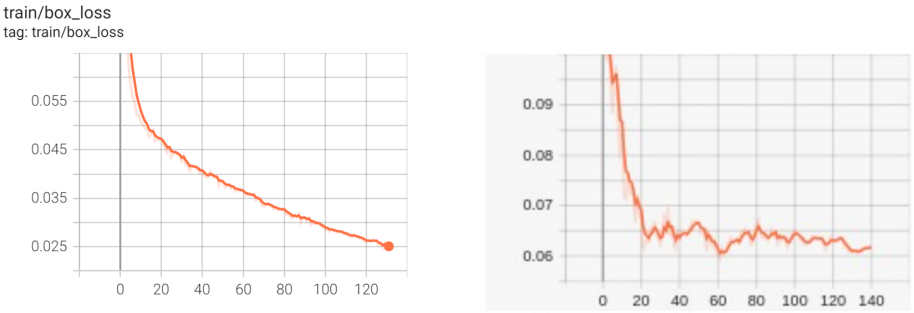

Figure 11. YOLOv5 training and validation graphs.

4.6. YOLOv6

With each iteration, the You Only Look Once model's goal was consistent: to rapidly learn how to predict bounding boxes for specific objects while preserving accuracy. The better a model is, the less hardware that is required to generate and operate it. YOLO models use an image as input and transmit it through a number of fully linked layers in the backbone. You only look at one model after the model uses the neck to represent these backbone elements. After receiving the neck features, the three heads of the YOLO models anticipate objectivity, class, and box reversion. In order to develop an efficient representation and representation-path integral-based convolutional (PAN) neck, YOLOv5 reconstructed the YOLO backbone and neck while accounting for the hardware. It had already been established that YOLOx and YOLOv6 had detachable heads, which implied the network had an additional layer separating these attributes from the ultimate head. Along with structural changes, the YOLOv5 repository also included several enhancements for the training procedure. These enhancements consisted of SIoU box regression loss, SimOTA tag assignment, and anchor-free (not NMS-free) training [33].

5. Results and Analysis

This part of the research discusses the results and observations described in the previous part of the research.

5.1. Dataset Validation Results

After collecting the exclusive dataset, the main task was to perform validation on the dataset and compare it with the public dataset. Here, the validation test was conducted on the exclusive dataset, as shown in Table 5, and then compared with the public datasets, as shown in Table 6.

Table 5. YOLOv5 on exclusive dataset.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
YOLOv5	0.61	1k	0.60	0.62	Exclusive Dataset

Table 6. YOLOv5 on public dataset.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
YOLOv5	0.60	1k	0.61	0.63	Public Dataset

The comparison chart between the private and public datasets is shown in Figure 12. Additionally, mAP (mean average precision) has established a benchmark for precision and recall in proprietary datasets, requiring the use of 2000 samples as training data.

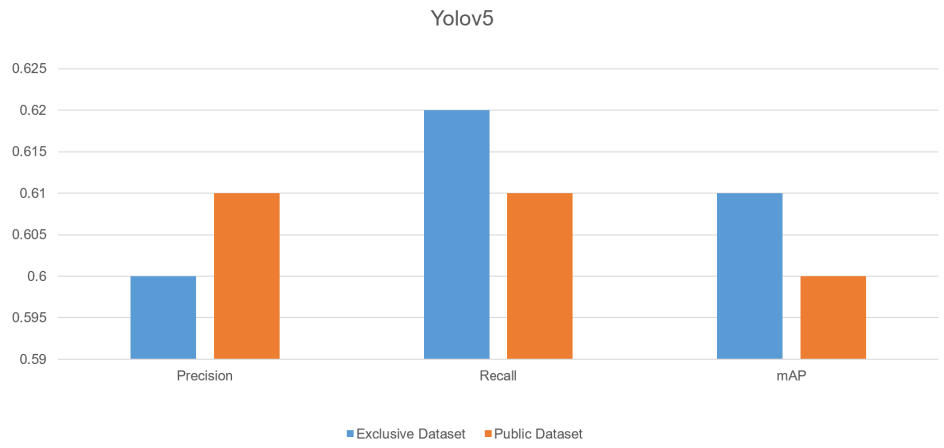

Figure 12. Dataset Validation Comparison Graph.

5.2. FasterRCNN

FasterRCNN has a considerable architecture. FasterRCNN is a fusion of Fast RCNN and region proposal, which makes algorithms very fast and accurate with low computational cost on hardware, such as CPU and GPU. We deployed the exclusive and public datasets on FasterRCNN architecture and achieved results, as shown in Table 7, which could be better than other existing studies.

Table 7. FasterRCNN results.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
FasterRCNN	0.45	2k	0.49	0.47	Custom, public dataset

Figure 13 shows the visual results of the model, and Figure 14 shows the precision-recall and mAP, which was 35%. This ratio was not optimal for object detection models.

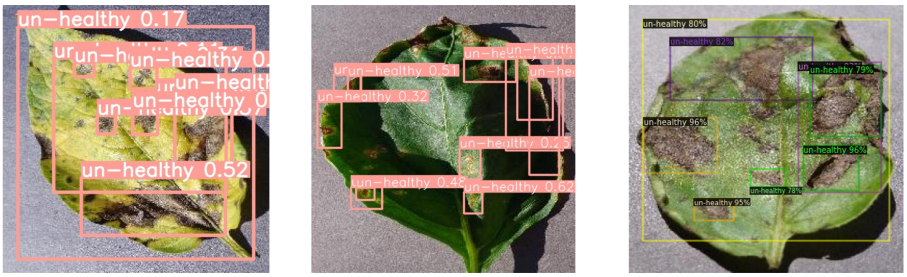

Figure 13. FasterRCNN visual representation.

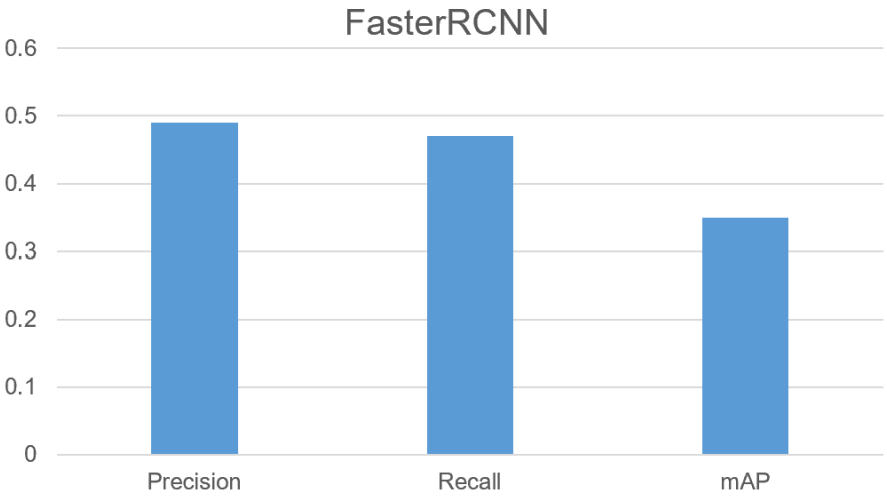

Figure 14. FasterRCNN comparison graph.

5.3. EfficentDET

As discussed above, EfficentDET architecture in implementations presents the visual representations of the results obtained from the EfficentDet code. We added the exclusive and public datasets on EfficentDet architecture with the same hyper-parameters, as shown in Figure 15. This was also not optimal, as compared to other existing studies. Figure 16 shows the visual results of EfficentDET.

```
#smells like some free compute from Colab, nice
gtf.Train_Dataset(root_dir, coco_dir, img_dir, set_dir, batch_size=8, image_size=512, use_gpu=True)
```

Figure 15. EfficentDET parameters.

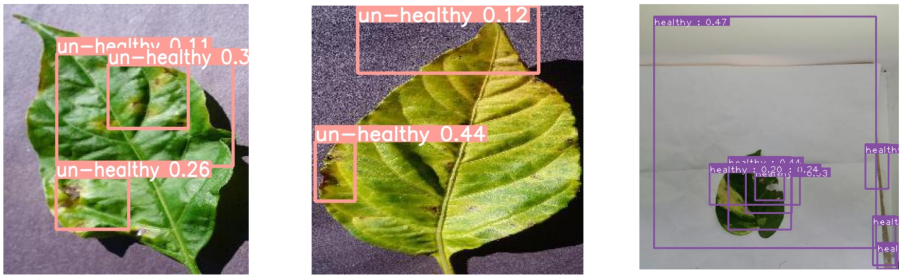

Figure 16. EfficentDET visual results.

Table 8 displays the EfficentDET findings and final results, which showed 35% accuracy with around 39% precision and 49% recall value.

Table 8. EfficentDET results.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
EfficentDET	0.35	2k	0.39	0.49	Custom, public dataset

Figure 17 depicts the average accuracy, which was about 60% on average. Though inferior to FasterRCNN, this was sufficient for an object detection model. Farming requires accurate detection while detecting objects. Many additional strategies were conducted, such as color correction and increasing epochs, but the accuracy required improvement. Hence, we switched to the best object detection model, YOLOv5.

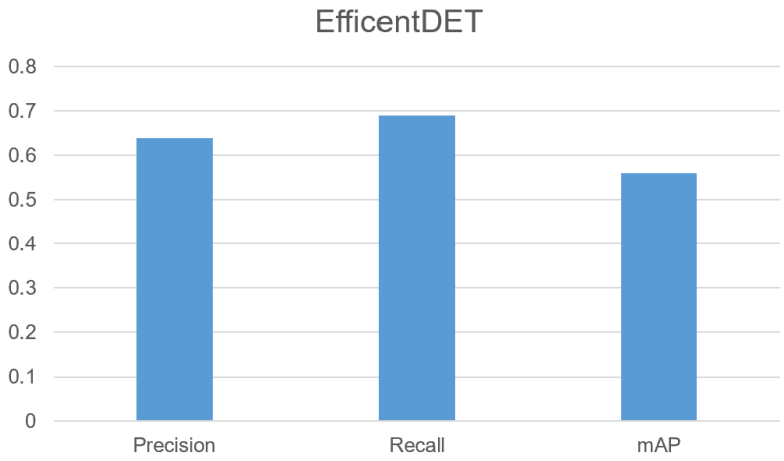

Figure 17. EfficentDET comparison graph.

5.4. YOLOv5 Experiments on Exclusive and Public Datasets

YOLOv5 is a state-of-the-art object recognition model that provides excellent mAP at low-resource demand. Firstly, YOLOv5 was applied to an exclusive dataset, but due to the few available datasets, we had to merge the exclusive dataset with the public to obtain minimum average precision. Figure 18 depicts the model summary, layers epochs, image size, etc.

Model summary: 213 layers, 7015519 parameters, 0 gradients, 15.8 GFLOPs

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 2/2 [00:01<00:00, 1.89it/s]
all	52	82	0.609	0.624	0.617	0.426
healthy	52	8	0.65	0.875	0.895	0.713
un-healthy	52	74	0.568	0.374	0.338	0.139

Accuracy 0.69%

Figure 18. YOLOv5 hyperparameters.

In the second experiment, we examined the validation loss, mAP, precision, and recall. Figure 19's graphs show the minimum average precision (mAP) at 0.5 and the minimum average precision in the range of 0.5 to 0.95. The criteria for accuracy, recall, and intersection over union (IoU) was used to plot the graphs.

$$Precision(P) = \frac{TP}{TP + FP} \tag{3}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{4}$$

- TP (True Positive) = How many instances were accurately detected

- *FP* (False Positive) = Number of incorrectly identified cases between healthy and unhealthy leaves
- *FN* (False Negative) = Number of unidentified cases between healthy and unhealthy leave
- IOU = Intersection over union
- K = threshold

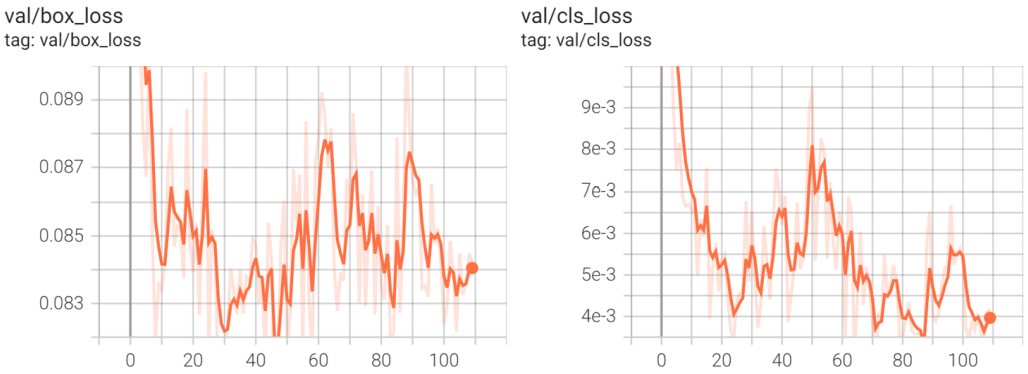

Figure 19. Validation loss representing a fluctuation in results due to insufficient datasets.

Graphs of mAP (mean average precision), precision, and recall for training data of 1000 samples are displayed in Figures 20 and 21. Graphs are an excellent tool for exploratory data analysis, as they provide an overview of the relationships throughout the whole dataset.

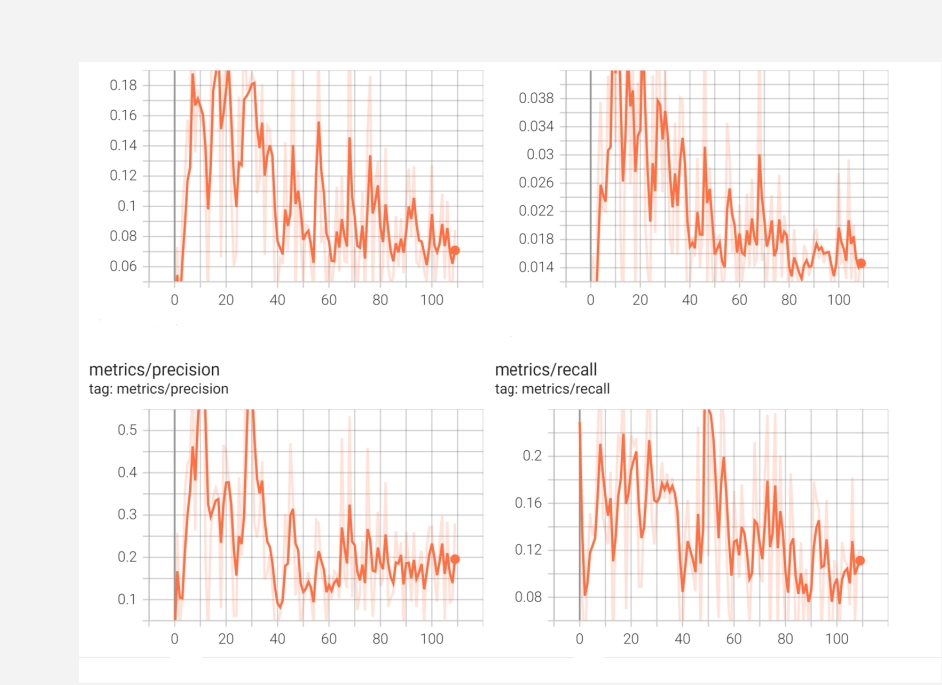

Figure 20. YOLOv5: mAP precision, and recall for training data of 1000 samples.

Figure 21. mAP Graphs mAP, precision, recall, is increasing or decreasing with changes in hyperparameters.

5.5. YOLOv5 Results

YOLO environment configuration with complete and public datasets is described below. To begin training, YOLOv5 YAML required two files. The first YAML defined the locations of the test and training data as well as the number of classes of objects being detected and the names of the items that belong to each class. The tuning parameters for training and testing are shown in Figure 22. The overall findings of our model are shown in Table 9, which were 93%, with a precision of 75% and a recall of 95%. The major advantage of YOLOv5 was that, as compared to FasterRcnn, YOLOv5 operates 2.5 times faster and managed better performance and detection of even small objects.

Table 9. YOLOv5 testing results.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
YOLOv5	0.93	2k	0.75	0.95	Custom, public dataset

```
!python train.py --img 416 --batch 16 --epochs 150 --data {dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Figure 22. YOLOv5 experiment 2.

Figure 23's graphs show the mAP at 0.5 and the mAP in the range of 0.5 to 0.95. The mAP graph was increasing incrementally with increasing epochs. The criteria for accuracy, recall, and intersection over union (IoU) were used to plot the graphs. Figure 23 depicts a precision graph that increases up to 25 epochs before fluctuating.

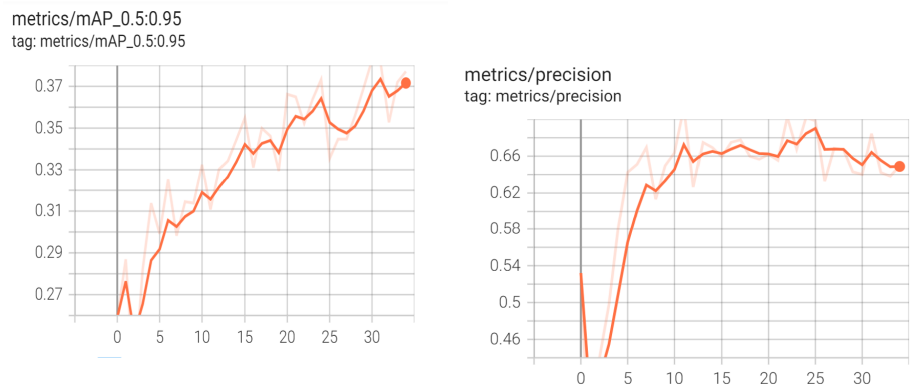

Figure 23. YOLOv5 training and validation graphs.

Figure 24 depicts the final recall metric, which was increasing with each epoch.

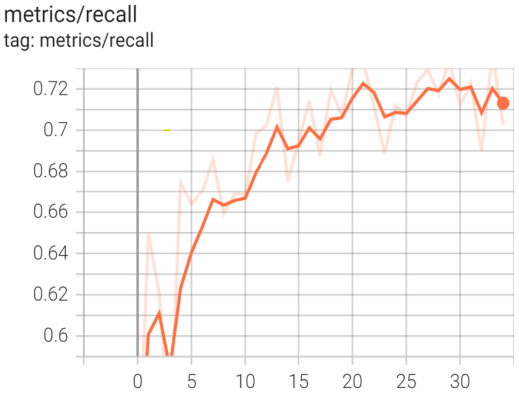

Figure 24. Recall graph YOLOv5: The recall is gradually increasing with the increment of epochs.

Figure 25 shows that the validation loss of YOLOv5 decreased significantly until epoch 20. After that, the validation loss declined and stopped at 0.06 and 0.05, at epoch 30.

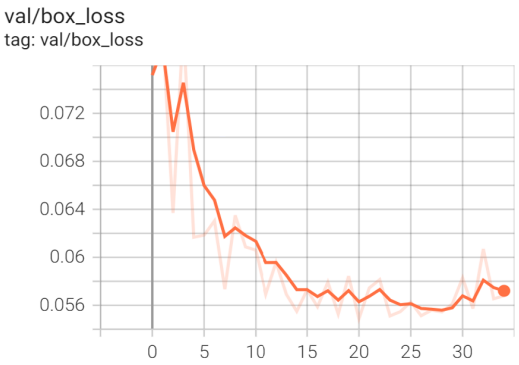

Figure 25. Validation loss graph YOLOv5: Validation loss was decreasing gradually, indicating the model prediction was significant.

Figure 26 shows the curve that indicated the confidence via F1 score.In the insight figure, the orange line shows the healthy part of detection, and the green color shows the unhealthy part of detection. The F1 curve indicated the PPV (precision) and TPR (recall) collectively as one visualization for every threshold.

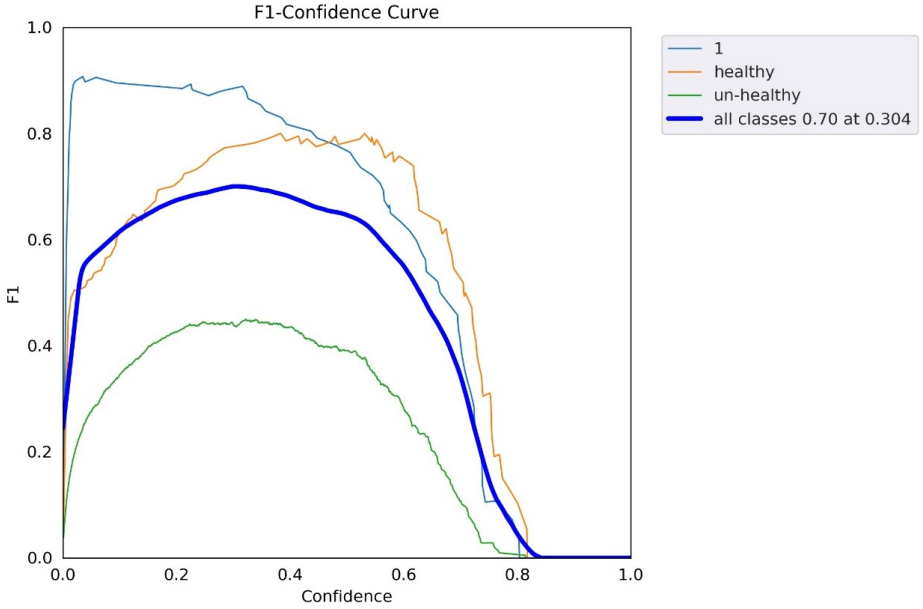

Figure 26. F1 confidence curve: Comparison of progression of mean F1 score across all experiments, grouped by training mechanism.

Figures 27 and 28 depict the confidence via recall and precision curves, respectively, using the R curve.

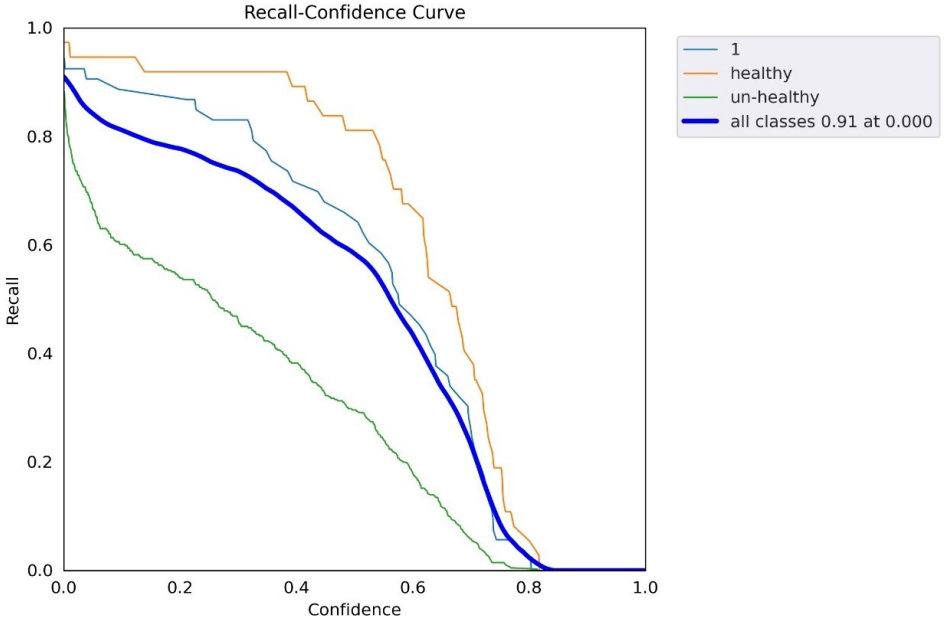

Figure 27. Confidence according to R Graph: Across all experiments, recall growth was compared and classified by training technique. It provided a substitute for the precision–recall curve.

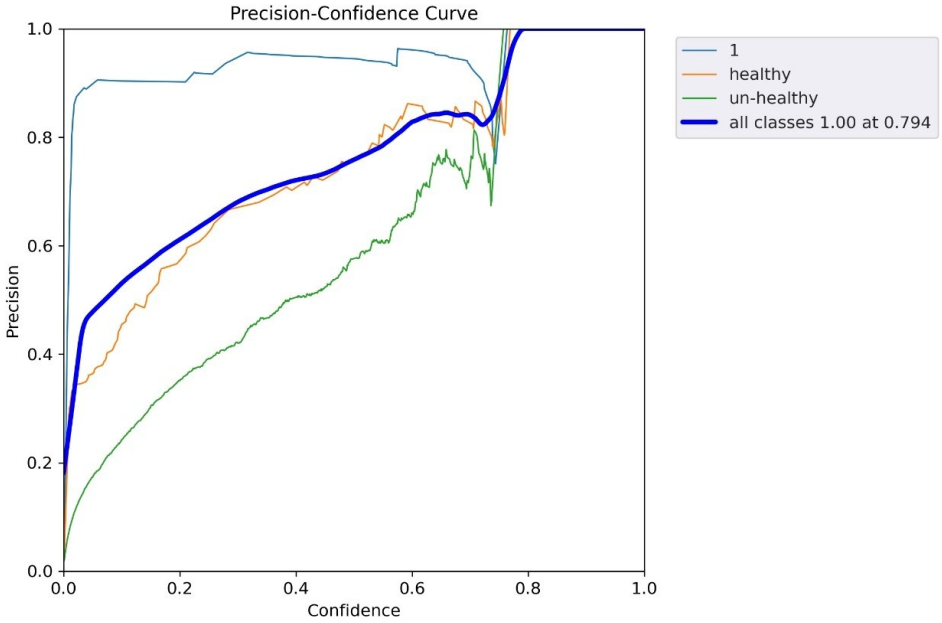

Figure 28. Confidence according to P curve graph: Comparison of the progression of precision across all experiments, grouped by training mechanism.

Figure 29 shows the compromise between recall and precision at different thresholds.

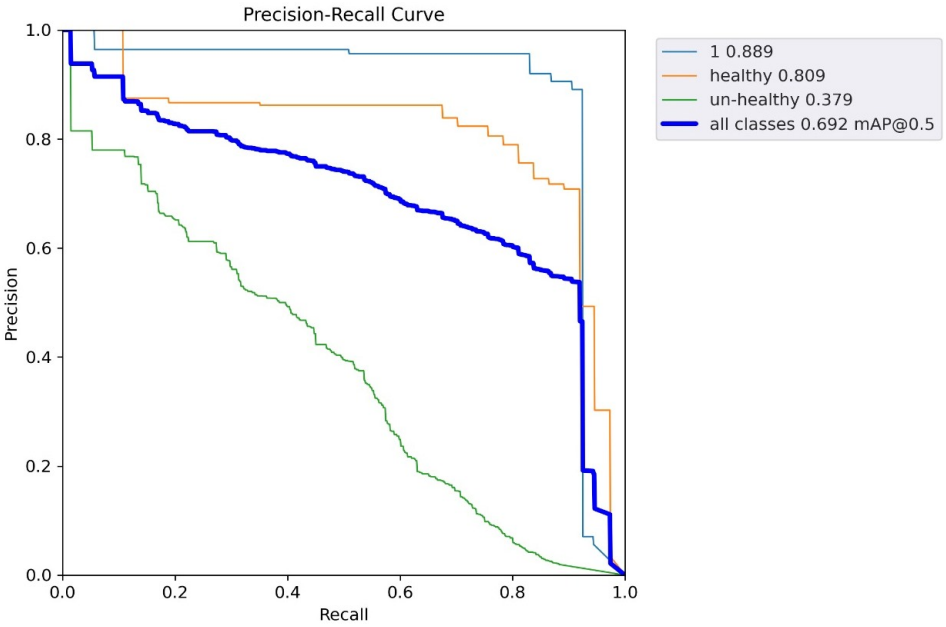

Figure 29. Precision according to recall graph:The precision via recall graph represents both high recall and high precision.

Confusion Matrix

A confusion matrix shows the variations between real and expected values, as shown in Figure 30. It assessed the effectiveness of our machine learning classification model using a table-like structure.

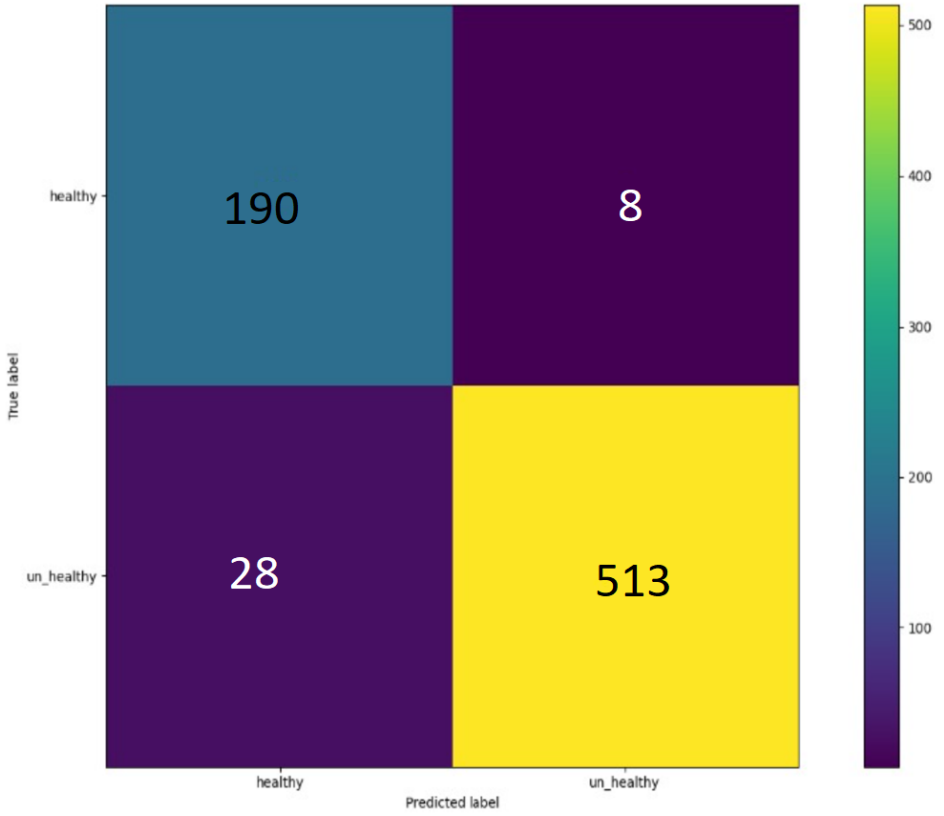

Figure 30. Confusion matrix for YOLOv5: healthy images (190) identified, 513 images were unhealthy and correctly identified.

A total of 190 images were true positives in this detection. Eight images in this selection were false positives, and 28 were false negatives. Furthermore, the last 513 images identified as unhealthy were correctly identified.

5.6. Experiments on YOLOv6

YOLOv6 was introduced recently with multiple changes. The same database was applied on YOLOv6, and results were obtained. As shown in Table 10, we deployed both datasets (exclusive and public datasets) on YOLOv6 and achieved 32% accuracy.

Table 10. YOLOv6 results.

Model Name	Accuracy	No. of Images	Precision	Recall	Dataset
YOLOv6	0.32	2k	0.35	0.58	Exclusive, Public dataset

Figure 31 shows a comparison graph that depicts precision 30%, recall, and mAP of the YOLOv6 model.

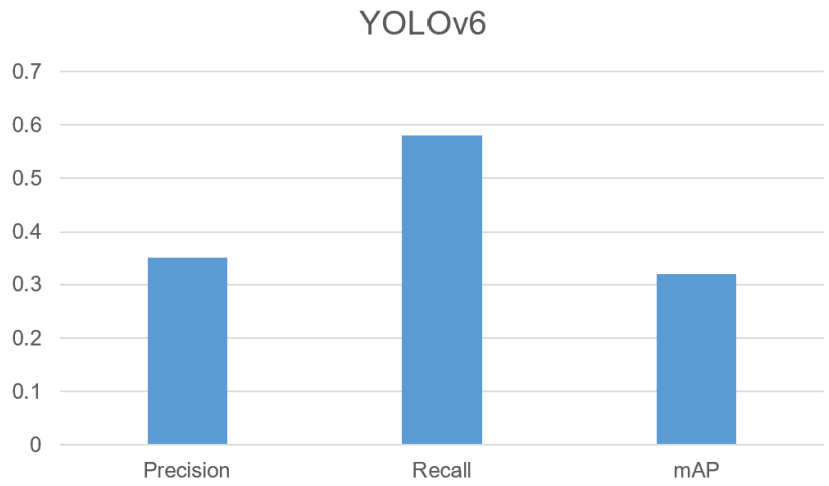

Figure 31. Comparison chart: Comparison between precision, recall, and mAP.

5.7. Comparative Analysis

Figure 32 shows a comparative analysis between different deep-learning algorithms (EfficentDet, FasterRCNN, YOLOv5, and YOLOv6). The graph displays the mAp score that was produced by FasterRCNN, YOLOv5 (our method), EfficentDet, and YOLOv6.

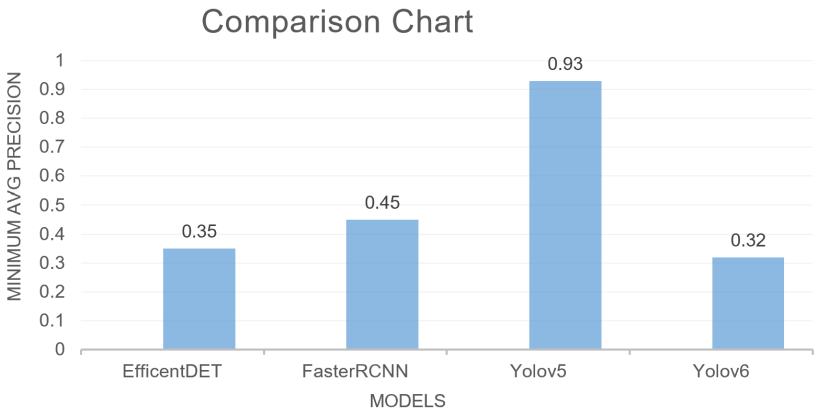

Figure 32. Comparison Chart: Comparative analysis between EfficentDET, FasterRCNN, YOLOv5, and YOLOv6.

6. Conclusions

This study used deep-learning techniques to classify healthy and unhealthy leaves. The identification and recognition of plant diseases in the ecological world are crucial for controlling plant diseases. In this research, a step-by-step procedure was performed. The first step of this research was gathering data; two types of datasets were included. These were the exclusive and public datasets, and then we performed the preprocessing steps on the datasets. Labeling was the most important step in preprocessing because this research had to follow a format acceptable to the selective neural network used for object detection and localization on a region of interest. Therefore, this research provided an acceptable format. Following preprocessing, this research employed augmentation techniques to increase the quantity and quality of the datasets. This research empirically compared four deep neural models to determine the best hyper-parameters and exclusive data validation. Based on FasterRCNN, the model had 0.49 precision with 0.47 recall, and the accuracy was 0.35, which was very low, as compared to the state-of-the-art models. Then, the dataset was deployed on the EfficentDET neural network, and the results were improved, as compared to FasterRCNN but not as good as state-of-the-art models. In this research, validation of the public dataset with the comparison of the exclusive dataset was also performed extensively.

Furthermore, after performing the validation test, the YOLOv5 model was trained on the public and exclusive datasets. Initially, the YOLOv5 model was trained on the pre-trained hyper-parameters, after which we adjusted the hyper-parameters, as shown in the results section, so the the mAP (0.5) was significant, and the final result was 93%. The approach presented in this research outperformed earlier iterations of YOLO in speed and accuracy. It may increase crop productivity by detecting and classifying plant disease.

7. Limitations

Our research had some limitations. There were still instances of missing or incorrect detection. In order to increase the model's detection precision, the model's mechanism needs to be further refined. Furthermore, using a high-resolution lenses for image capture could improve accuracy further.

Future Work

After completing the work described above (e.g., dataset collection, preprocessing, data annotation, data validation, and empirical investigation), we showed that fast detection is possible, but it still requires specific hardware designs. Certain gaps need to be filled by future research. The accuracy, execution time, and minimum average precision of the models should be higher if the dataset is gathered using high-quality lenses (ultra-wide angle) and a large team (7–8 members). YOLOv5 architecture will be improved in the future and deployed as an android application, so it can be used for real-time object recognition with the assistance of YOLOv5's improved architecture.

Author Contributions: Conceptualization, M.K., U.I., M.U.A., M.S.S., G.N. and H.T.R.; methodology, M.K.; validation, U.I., M.U.A., M.S.S., G.N. and H.T.R.; formal analysis, U.I. and M.U.A.; investigation, U.I. and M.U.A.; data curation, M.K. and M.S.S.; writing—original draft preparation, M.K.; writing—review and editing, M.S.S., U.I., M.U.A., G.N. and H.T.R.; supervision, M.S.S., G.N. and H.T.R.; Resources, M.S.S., G.N. and H.T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research has not received any funding.

Data Availability Statement: The dataset shall be available through declaration from all authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
2. Deng, L.; Deng, Q. The basic roles of indoor plants in human health and comfort. *Environ. Sci. Pollut. Res.* **2018**, *25*, 36087–36101. [CrossRef]
3. Balasundram, S.K.; Golhani, K.; Shamshiri, R.R.; Vadamalai, G. Precision agriculture technologies for management of plant diseases. In *Plant Disease Management Strategies for Sustainable Agriculture through Traditional and Modern Approaches*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 259–278.
4. Trivedi, P.; Leach, J.E.; Tringe, S.G.; Sa, T.; Singh, B.K. Plant–microbiome interactions: From community assembly to plant health. *Nat. Rev. Microbiol.* **2020**, *18*, 607–621. [CrossRef] [PubMed]
5. Wang, X.; Yang, W.; Wheaton, A.; Cooley, N.; Moran, B. Efficient registration of optical and IR images for automatic plant water stress assessment. *Comput. Electron. Agric.* **2010**, *74*, 230–237. [CrossRef]
6. Khan, S.; Narvekar, M.; Hasan, M.; Charolia, A.; Khan, A. Image processing based application of thermal imaging for monitoring stress detection in tomato plants. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; IEEE: New York, NY, USA, 2019; pp. 1111–1116.
7. Hasan, R.I.; Yusuf, S.M.; Alzubaidi, L. Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants* **2020**, *9*, 1302. [CrossRef] [PubMed]
8. Arivazhagan, S.; Shebiah, R.N.; Ananthi, S.; Varthini, S.V. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agric. Eng. Int. CIGR J.* **2013**, *15*, 211–217.
9. Lin, K.; Gong, L.; Huang, Y.; Liu, C.; Pan, J. Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Front. Plant Sci.* **2019**, *10*, 155. [CrossRef]

10. Blumenthal, J.; Megherbi, D.B.; Lussier, R. Supervised machine learning via Hidden Markov Models for accurate classification of plant stress levels & types based on imaged Chlorophyll fluorescence profiles & their rate of change in time. In Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), Annecy, France, 26–28 June 2017; IEEE: New York, NY, USA, 2017; pp. 211–216.
11. Shrivastava, V.K.; Pradhan, M.K. Rice plant disease classification using color features: A machine learning paradigm. *J. Plant Pathol.* **2021**, *103*, 17–26. [CrossRef]
12. Chen, M.; Tang, Y.; Zou, X.; Huang, K.; Huang, Z.; Zhou, H.; Wang, C.; Lian, G. Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Comput. Electron. Agric.* **2020**, *174*, 105508. [CrossRef]
13. Li, Q.; Jia, W.; Sun, M.; Hou, S.; Zheng, Y. A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment. *Comput. Electron. Agric.* **2021**, *180*, 105900. [CrossRef]
14. Mishra, M.; Choudhury, P.; Pati, B. Modified ride-NN optimizer for the IoT based plant disease detection. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 691–703. [CrossRef]
15. Abisha, A.; Bharathi, N. Review on Plant health and Stress with various AI techniques and Big data. In Proceedings of the 2021 International Conference on System, Computation, Automation and Networking (ICSCAN), Puducherry, India, 30–31 July 2021; IEEE: New York, NY, USA, 2021; pp. 1–6.
16. Chandel, N.S.; Chakraborty, S.K.; Rajwade, Y.A.; Dubey, K.; Tiwari, M.K.; Jat, D. Identifying crop water stress using deep-learning models. *Neural Comput. Appl.* **2021**, *33*, 5353–5367. [CrossRef]
17. Jiang, B.; Wang, P.; Zhuang, S.; Li, M.; Gong, Z. Drought stress detection in the middle growth stage of maize based on gabor filter and deep learning. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; IEEE: New York, NY, USA, 2019; pp. 7751–7756.
18. Ahmed, F.; Al-Mamun, H.A.; Bari, A.H.; Hossain, E.; Kwan, P. Classification of crops and weeds from digital images: A support vector machine approach. *Crop Prot.* **2012**, *40*, 98–104. [CrossRef]
19. Abdulridha, J.; Ehsani, R.; Abd-Elrahman, A.; Ampatzidis, Y. A remote sensing technique for detecting laurel wilt disease in avocado in presence of other biotic and abiotic stresses. *Comput. Electron. Agric.* **2019**, *156*, 549–557. [CrossRef]
20. Khan, R.U.; Khan, K.; Albattah, W.; Qamar, A.M. Image-based detection of plant diseases: From classical machine learning to deep learning journey. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5541859. [CrossRef]
21. 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, Hyderabad, India, 5–7 January 2020; pp. 249–253.
22. Mathew, M.P.; Mahesh, T.Y. Leaf-based disease detection in bell pepper plant using YOLOv5. *Signal Image Video Process.* **2022**, *16*, 841–847. [CrossRef]
23. Khan, S.; Tufail, M.; Khan, M.T.; Khan, Z.A.; Anwar, S. Deep learning-based identification system of weeds and crops in strawberry and pea fields for a precision agriculture sprayer. *Precis. Agric.* **2021**, *22*, 1711–1727. [CrossRef]
24. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13029–13038.
25. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Paez, A.; Gebre, G.M.; Gonzalez, M.E.; Tschaplinski, T.J. Growth, soluble carbohydrates, and aloin concentration of Aloe vera plants exposed to three irradiance levels. *Environ. Exp. Bot.* **2000**, *44*, 133–139. [CrossRef] [PubMed]
27. Jajja, A.I.; Abbas, A.; Khattak, H.A.; Niedbała, G.; Khalid, A.; Rauf, H.T.; Kujawa, S. Compact Convolutional Transformer (CCT)-Based Approach for Whitefly Attack Detection in Cotton Crops. *Agriculture* **2022**, *12*, 1529. [CrossRef]
28. Niedbała, G.; Kurek, J.; Świderski, B.; Wojciechowski, T.; Antoniuk, I.; Bobran, K. Prediction of Blueberry (*Vaccinium corymbosum* L.) Yield Based on Artificial Intelligence Methods. *Agriculture* **2022**, *12*, 2089. [CrossRef]
29. Jasim, M.A.; Al-Tuwaijari, J.M. Plant leaf diseases detection and classification using image processing and deep-learning techniques. In Proceedings of the 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 16–18 April 2020; IEEE: New York, NY, USA, 2020; pp. 259–265.
30. Swain, S.; Nayak, S.K.; Barik, S.S. A review on plant leaf diseases detection and classification based on machine learning models. *Mukt Shabd* **2020**, *9*, 5195–5205.
31. Ranjan, M.; Weginwar, M.R.; Joshi, N.; Ingole, A. Detection and classification of leaf disease using artificial neural network. *Int. J. Tech. Res. Appl.* **2015**, *3*, 331–333.
32. Bolliger, P.; Ostermaier, B. Koubachi: A mobile phone widget to enable affective communication with indoor plants. In Proceedings of the Mobile Interaction with the Real World (MIRW 2007), Singapore, 9 September 2007; p. 63.
33. Gélard, W.; Herbulot, A.; Devy, M.; Casadebaig, P. 3D leaf tracking for plant growth monitoring. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; IEEE: New York, NY, USA, 2018; pp. 3663–3667.
34. Mishra, P.; Feller, T.; Schmuck, M.; Nicol, A.; Nordon, A. Early detection of drought stress in Arabidopsis thaliana utilsing a portable hyperspectral imaging setup. In Proceedings of the 2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.

35. Alexandridis, T.K.; Moshou, D.; Pantazi, X.E.; Tamouridou, A.A.; Kozhukh, D.; Castef, F.; Lagopodi, A.; Zartaloudis, Z.; Mourelatos, S.; de Santos, F.J.N.; et al. Olive trees stress detection using Sentinel-2 images. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: New York, NY, USA, 2019; pp. 7220–7223.
36. Ciężkowski, W.; Kleniewska, M.; Chormański, J. Using Landsat 8 Images for The Wetland Water Stress Calculation: Upper Biebrza Case Study. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; IEEE: New York, NY, USA, 2019; pp. 6867–6870.
37. Bhugra, S.; Chaudhury, S.; Lall, B. Use of leaf colour for drought stress analysis in rice. In Proceedings of the 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Patna, India, 16–19 December 2015; IEEE: New York, NY, USA, 2015; pp. 1–4.
38. Ahmed, K.; Shahidi, T.R.; Alam, S.M.I.; Momen, S. Rice leaf disease detection using machine learning techniques. In Proceedings of the 2019 International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 24–25 December 2019; IEEE: New York, NY, USA, 2019; pp. 1–5.
39. Zhang, K.; Wu, Q.; Liu, A.; Meng, X. Can deep learning identify tomato leaf disease? *Adv. Multimed.* **2018**, *2018*, 6710865. [CrossRef]
40. Selvaraj, M.G.; Vergara, A.; Ruiz, H.; Safari, N.; Elayabalan, S.; Ocimati, W.; Blomme, G. AI-powered banana diseases and pest detection. *Plant Methods* **2019**, *15*, 92. [CrossRef]