Khan, Asif Iqbal; Quadri, S.M.K; Banday, Saba et al.

Working Paper

Deep Diagnosis : A Real-Time Apple Leaf Disease Detection System Based on Deep Learning

Provided in Cooperation with:
Social Science Research Network (SSRN)

Reference: Khan, Asif Iqbal/Quadri, S.M.K et. al. (2022). Deep Diagnosis : A Real-Time Apple Leaf Disease Detection System Based on Deep Learning. [S.l.] : SSRN.
https://ssrn.com/abstract=4019467.
https://doi.org/10.2139/ssrn.4019467.
doi:10.2139/ssrn.4019467.

This Version is available at:
http://hdl.handle.net/11159/503114

Title: Deep Diagnosis: A Real-Time Apple Leaf Disease Detection System Based on Deep Learning

Authors:

1) **Asif Iqbal Khan** (Corresponding author)
   Dept. of Computer Science, Jamia Millia Islamia, New Delhi
   Email: khanasifiqbal7@gmail.com
   Mb: +91-9858323226

2) S.M.K Quadri
   Head, Dept. of Computer Science, Jamia Millia Islamia, New Delhi
   Email: quadrismk@jmi.ac.in

3) Saba Banday
   Dept. of Pathology, Sher-i-Kashmir University of Agricultural Sciences, Shalimar
   Email: sababanday@gmail.com

4) Junaid Latief Shah
   Higher Education Department, J&K, India
   Email: junaidlatiefshah@gmail.com

Abstract

Background and Objective: Diseases and pests are one of the major reasons for low productivity of apples which in turn results in huge economic loss to the apple industry every year. Early detection of apple diseases can help in controlling the spread of infections and ensure better productivity. However, early diagnosis and identification of diseases is challenging due to many factors like, presence of multiple symptoms on same leaf, non-homogeneous background, differences in leaf colour due to age of infected cells, varying disease spot sizes etc. This paper is an attempt to overcome these challenges and provide apple disease detection and identification system that can perform real-time detection of these diseases with higher accuracy and faster detection speed.

Methods: In this study, we first constructed an expert-annotated apple disease dataset of suitable size consisting around 9000 high quality RGB images covering all the main foliar diseases and symptoms. Next, we propose a deep learning based apple disease detection system which can efficiently and accurately identify the symptoms. The proposed system works in two stages, first stage is a tailor-made light weight classification model which classifies the input images into diseased, healthy or damaged categories and the second stage (detection stage) processing starts only if any disease is detected in first stage. Detection stage performs the actual detection and localization of each symptom from diseased leaf images. Both classification and detection models were initialized using transfer learning and then trained on prepared dataset. Data augmentation techniques like rotation, translation, noise, cut-out etc. were also applied to prevent overfitting.

Results: The proposed approach obtained encouraging results, reaching around 88% of classification accuracy and our best detection model achieved mAP of 42%. The qualitative results validate that the proposed system is effective in detecting various types of apple diseases and can be used as a practical tool by farmers. The preliminary results of this study look promising even on small or tiny spots.

Conclusion: The proposed apple disease detection system achieved promising results and can be very helpful tool for farmers and apple growers to aid them in diagnosis, quantification and follow-up of infections. Furthermore, in future, the work can be extended to other fruits and vegetables as well.

Keywords: Deep Learning, Apple leaf disease detection, Convolutional Neural Network. Object detection

1. INTRODUCTION

Apple fruits are cultivated worldwide and are one of the most widely grown fruits in the world. Apple fruit is one of the most productive fruit due to its high remedial and nutritive values. In India, the valley of Kashmir holds the maximum share of Apple production with more than 75% of total apple production. Currently, around 160000 hectares of land in the Valley is under apple cultivation with an annual productivity of around 180000 MTs (Directorate of Horticulture, 2021), in which an enormous proportion is exported to different parts of the world. However, diseases and pests cause huge economic loss to the apple industry every year. Diseases like Alternaria, Scab, and Mosaic remain a major threat for the apple growers. In July 2013, Alternaria unfurled in apple orchards and spread like wildfire infecting more than 70% of the cultivars in many districts across Valley. The disease resulted in extensive fruit fall and hence decrease in fruit production (Bhat et al., 2015). The same disease attack was again reported in 2018 (Brahimi et al., 2017). As per the domain experts and scientists, one of the main reasons for the spread of this catastrophic disease was the lack of proper disease forecasting and detection system (Bhat et al., 2015). In 2020, a severe scab infection broke out and more than 30 percent crop was affected by diseases despite the use of fungicides. The infection resulted in heavy wastage of crop (Raashid Hassan, 2021).

An on-time detection system would detect the disease and prevent the widespread of disease among other plants, thus preventing substantial economic losses. Pests and diseases not only reduce the production but adversely affect the fruit quality as well. Therefore, timely detection of diseases is crucial for enhancing both quality and quantity of apples. Identifying a disease correctly when it first appears would help the farmer to take proper precautionary measures or apply just the right amounts of pesticides, thereby getting both economic and environmental benefits.

Traditionally, plant disease diagnosis was done by domain experts by visually examining the leaves. However, this practice was labour intensive and time-consuming (Dutot et al., 2013). Also, the experts must be proficient and should have extensive knowledge of various diseases, their symptoms and treatment. A satisfactory alternative to such labour intensive task will be an automatic detection system that can detect plant diseases in early stages. In this context, over the years, various computer vision and image processing techniques have been studied and developed to detect and identify plant diseases. Dubey and Jalal, 2012 introduced a technique for classification of apple fruit diseases. They first used K-Means clustering for defect segmentation followed by Multi-class Support Vector Machine (SVM) for classification. Features like Global Color Histogram (GCH), Color Coherence Vector (CCV), Local Binary Pattern (LBP) and Completed

Local Binary Pattern (CLBP) were used for classification. Mokhtar et al, 2015, proposed a technique to detect tomato diseases based on SVM. Dandawate and Kokare 2015 and Raza et al, 2015 also used SVM based approaches for classification of plant diseases. Chuanlei et al., 2017 proposed a pattern recognition based technique which used region growing algorithm (RGA) to detect apple leaf diseases and achieved an accuracy of 90% on a database of 90 disease images.

Deep Learning is popular Machine Learning sub field, which has gained popularity in recent years. Deep Learning algorithms are a special type of Artificial Neural Networks which extract high-level representations of data while training without any human intervention. Deep learning has recently solved many complex problems and has shown excellent performance in many computer vision and machine learning tasks like image classification, object detection, speech recognition, voice recognition, natural language processing, medical imaging etc. (Abbas et al., 2021). Due to their ability to learn features directly from images, deep learning algorithms like Convolutional Neural Networks (CNN) have found their application in the field of Agriculture as well. It is now being widely used in plant disease identification and diagnosis. For example apples (Liu et al., 2018), bananas (Amara et al., 2017), cucumber (Kawasaki et al., 2015), tomato (Brahimi et al., 2017), Multiple Plant diseases (Prasanna *et al., 2016), cassava (*Ramcharan et al., 2017*).*

In 2017, Brahimi et al, introduced deep learning for classifying tomato disease based on leaf images and achieved classification accuracy of up to 99%, easily outperforming the conventional methods. Ferentinos, (2018) trained different CNN architectures (AlexNet, VGG and GoogLeNet), on an open database containing 58 categories of plants or diseases. According to the experimental results, VGG architecture achieved the best accuracy of 99.53%. Liu et al. (2018) proposed a deep model based on AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015), for classification of apple leaf diseases. Zhang et al., 2019 proposed a deep convolutional neural network model (DCNN) for apple disease classification. The proposed model used one global average pooling layer instead of all the full connection layers and utilized modified Softmax for improvement. Jiang et al. (2019) used object detection algorithm called SSD with Inception module and Rainbow concatenation to detect 5 types of apple diseases. The model was trained on a hybrid dataset (artificially generated and collected from field) but the dataset is not publically available for testing. Zhong et al. (2020) proposed DenseNet-121 based approach for multi-classification of apple diseases and achieved an accuracy of 93.1%. However, the results are obtained on a small test set and contains only 3 diseases divided into 5 classes based on the intensity of the disease.

All the models and approaches proposed for apple disease identification discussed above have one common weakness, they all have been tested on single symptom images i:e the images in the dataset contain only one type of disease. Whereas in reality, more often than not, a leaf can be infected with multiple diseases and varying intensities. Hence, there remains a lack of performance in real-field scenarios.

Challenges

Plant disease detection involves many challenges and complexities due to various factors and in this research, we are trying to overcome these challenges and propose a real-time and accurate End-to-End apple disease detection system.

Plant disease patterns vary with the season and other factors such as CO2 concentration, humidity, temperature, and water availability. These factors can remarkably affect the disease development as each disease may respond differently to these variations (Velasquez et al., 2018). Moreover, disease patterns significantly vary (high intra-class variation) due to factors like leaf morphology, non-homogeneous background, age of infected cells, differences in leaf colour and light illumination during imaging, and on the other hand, sometimes the visual symptoms of different diseases may appear similar (low inter-class variation) due to varying lighting conditions etc.

Next challenge that we are dealing with is the detection of small spots during early stages. Detecting small objects is a difficult task and most of the object detection algorithms have tough time in dealing with small objects (Nguyen et al. 2020). Disease spots come in a wide variety of shapes, sizes and colours and occur randomly on the leaf surface. In incipient form, these spots are usually small or even tiny and with time they change colour, size and shape making it difficult for traditional detection algorithms.

Another challenge is the availability of apple leaf diseases dataset. Deep learning models are data hungry models which require huge amount of data for training. To the best of our knowledge, there is no sufficient sized large scale dataset available that can be utilized for this research. There are few apple leaf disease classification datasets available online but those datasets cover only few diseases and since the datasets are not annotated, those are not suitable for this research. So we built a sufficient sized quality dataset covering all the major apple diseases and pests. Therefore, the contribution of this research is twofold.

a) First, we have created an expert-annotated apple disease dataset of suitable size consisting of high quality RGB images covering all the main foliar diseases and symptoms. A good quality dataset is the first important requirement for this research. Since there is no suitable dataset available right now, our first aim was to build a suitable dataset which can be used to train and test our models. The dataset contains diseased apple leaf images with different backgrounds collected from various fields/orchards under different lighting conditions using different capturing devices.

b) Second, we present a deep learning based real-time leaf disease detection system to identify seven types of diseases and pests that affect apple plants which can help fruit growers in accurately identifying various disease on time and provide helpful recommendations.

2. DATASET

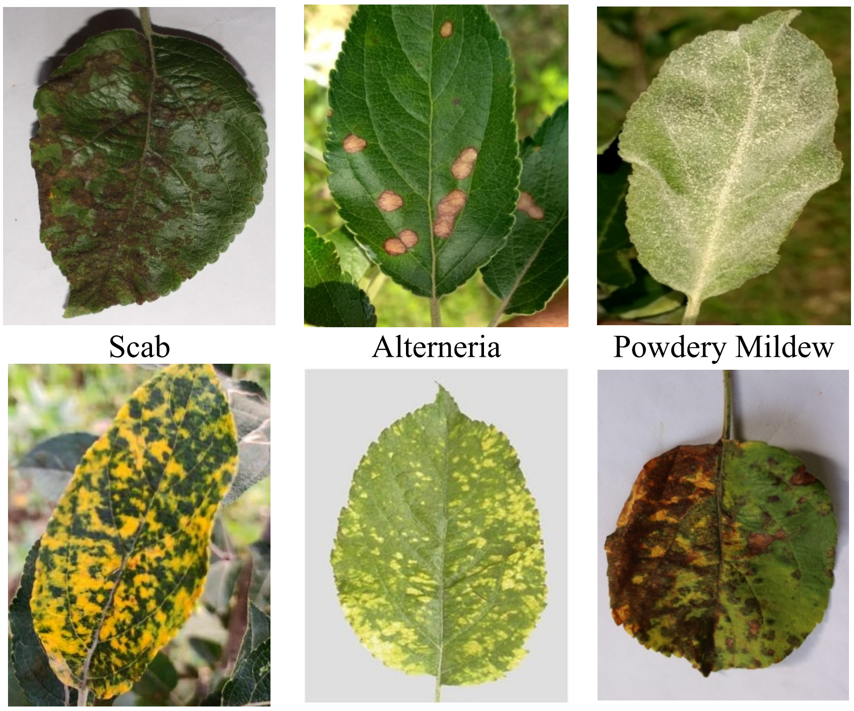
Apple Leaf Disease Dataset

The dataset preparation consists of two parts, i) *data collection*: collection of diseased apple leaf images from various fields and orchards and ii) *data annotation*: labelling of images as per their symptoms. Both these tasks are time consuming and resource intensive.

Data Collection: Our dataset preparation took huge amount of human efforts as well as material resources. First it took many human resources to collect the images of diseased apple leaves visiting various fields and orchards in different seasons. The total data collection took around two years to complete. There are around seven varieties of apples and eight types of diseases and pests commonly found in the valley. For this study, we have considered five commonly found diseases viz. Scab, Alternaria, Apple Mosaic, Marssonina leaf blotch (MLB) and powdery mildew only. We also made sure that our dataset contains leaf images from each variety. Not only this, disease patterns change with time and other factors like temperature, rainfall, humidity etc. Some diseases only appear in a particular season and time, for example rainy and humid weather is favourable for the development of Marsonina Leaf Blotch (MLB). Therefore, the disease appears very late towards the end of July and is more noticeable in August. As a result, a lot of time and efforts went into waiting and monitoring those diseases.

Apart from the diseases mentioned above, we have also considered identifying other visual symptoms as well. For example insect attack and Necrosis. Necrosis means death of body cells or tissues. It is not a disease but a symptom of disease or stress due to injury, radiation or chemicals. The symptom can be dark watery spots or sometimes tan or black papery spots on leaves. Portions of the plant may appear yellow or wilted, indicating a systemic severe disease that leads to death of cells. The most common causes of leaf Necrosis are diseases like alternaria, mossaic etc. However, weather-related problems, insect activity and nutritional deficiency may also result in death of leaf cells or tissues.

We collected more than 6000 leaf images in two years. To reflect the real world scenario, we took pictures from different cameras including mobile phones of different brands. The images have non-homogeneous background taken at different disease maturity stage and under different lighting conditions. Figure 1 shows some of the sample images from our collected dataset. After image capturing, we performed manual inspection of these images and removed all duplicate, low quality, damaged and those images which were beyond recognition due to severe disease attack. Finally we were left with 5201 images ready for annotation out of which 2739 images had single symptom/disease and 2462 images had multiple symptoms. Table I lists the class wise breakup of images.


Scab Alternaria Powdery Mildew

MLB Apple Mosaic Multiple

Figure 1: Sample images from the prepared Dataset

Table I: # of images of each disease in our apple disease dataset

Disease/Symptoms	Number of Images
Scab only	639
Alternaria only	418
Mossaic only	312
MLB only	371
Powdery Mildew only	320
Necrosis only	125
Insect only	221
Multiple symptoms	2462
Healthy	221
Damaged	112
Total	**5201**

Data Annotation: Data annotation is a vital step in the success of the machine learning models. In data annotation, each spot/object in an image is manually labelled with its tag or label. There are several tools available for image annotation and in this study we used LabelImg (Tzutalin, 2015). LabelImg is a graphical image annotation tool written in python which generates annotations as XML files in PASCAL VOC format. Besides, it also supports YOLO and CreateML formats (Tzutalin, 2015). This labelled data is used by supervised machine learning algorithms during training.

Data annotation is a manual process thus requires human resources, time and domain knowledge. Any fault or error in labelling can lead to lower success rates of the model. Domain knowledge is important for labelling and through data labelling, the knowledge of a Subject Matter Experts (domain experts) with years of experience is instilled into machine learning models. Therefore, for this study, data labelling was performed under the supervision of two domain experts who monitored the whole process and finally the dataset was used only after each annotation was cross checked by these experts.

Data Augmentation: Data Augmentations are needed for supplementing and enriching the training data. After collecting and annotating around 5000 images, various data augmentation operations were applied to make the dataset richer and diverse. This helps in increasing the generalizability of deep models and overcome the problem of overfitting. With more images and diversity achieved by data augmentation techniques, the model can learn as many relevant features as possible during the training process, thus avoiding overfitting and achieving higher performance. There are many augmentation options available and the ones we used are

i) *Rotation*: randomly rotate images clockwise or anti-clockwise up to a certain degree amount.

ii) *Flip*: Randomly flip the images vertically or horizontally.

iii) *Noise*: Randomly inject noise (salt and pepper) to some images.

iv) *Brightness:* Increase/decrease brightness of random images by certain amount.

v) *Exposure:* Adjust the gamma exposure (darker or brighter) of an image up to certain amount.

vi) *Cut-out:* Removes sections of an image to simulate occlusion.

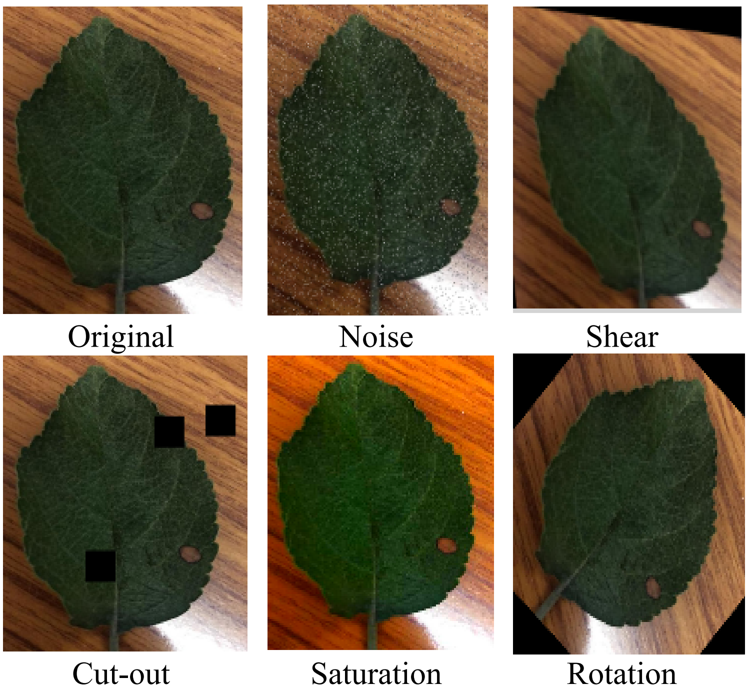


Figure 2: Effect of different augmentation operations on original image

Apart from theses image level augmentation, we also applied few bounding box level augmentation operations as well. To visualize how these transformations affect the original images, a plot showing all the transformations next to the original image for comparison is shown in Figure 2.

After applying data augmentation, our final dataset comprises of 9000 images fully annotated with 9 classes. The prepared dataset is first of its kind in terms of the size and scale. A sufficient sized dataset not only helps the current study but would potentially be helpful for future research in the area.

3. METHODOLOGY

Object detection is a complex task due to many challenges and disease detection from leaf images comes with its own set of challenges. To overcome such challenges we divided the overall disease detection process into two main stages. These two stages are a) Initial Classification and b) Detection stage. The overall pipeline is shown in Figure 3 and overall structure of our detection system is presented in Figure 4. The Initial Classification stage first classifies the input image into one of the following ten categories

i) Healthy: If the leaf is healthy and free from any disease.

ii) Scab: If the leaf is infected with Scab only.

iii) Alternaria: If the leaf is infected with Alternaria only.

iv) Powdery-Mildew: If the leaf is infected with Powdery-Mildew only.

v) Mossaic: If the leaf is infected with Mossaic virus only.

vi) MLB: If the leaf is infected with Marsonina Leaf blotch only.

vii) Multiple: If the leaf is infected with multiple diseases i:e the leaf has multiple visual symptoms on it.

viii) Necrosis: If the leaf has dead tissues.

ix) Insect: If the leaf is attacked by insect/s.

x) Damaged: If the leaf or major portion of the leaf is damaged due to severe disease intensity rendering it unrecognizable.

If the input image is classified as ‘Healthy’ or ‘Damaged’ the process stops without going further otherwise the output is fed to the Detection stage. The reason for using a classification stage and not using the detection stage directly is to improve the performance and reduce overall processing time as overlapping symptoms deform a leaf area resulting in wrong detections. Detection stage processing would start only if any disease is detected in first stage. This way all the invalid inputs (a healthy leaf image with no visual symptoms or severely damaged leaf) are filtered out. There is no need to go through the detection phase if there are no symptoms at all or the leaf area is severely damaged and cannot be diagnosed. A drill down on individual stages is discussed below.

3.1. The Initial Classification Stage

The “Initial Classification stage’ of our proposed system consists of a classification model which classifies an input image into one of the 10 classes discussed above. The classification model is a light weight Convolutional Neural Network architecture tailored for apple disease classification. It is based on Xception architecture (Chollet, 2017). Xception stands for Extreme version of Inception (Szegedy et al., 2016) (its predecessor model) is a 71 layers deep CNN architecture pre-trained on

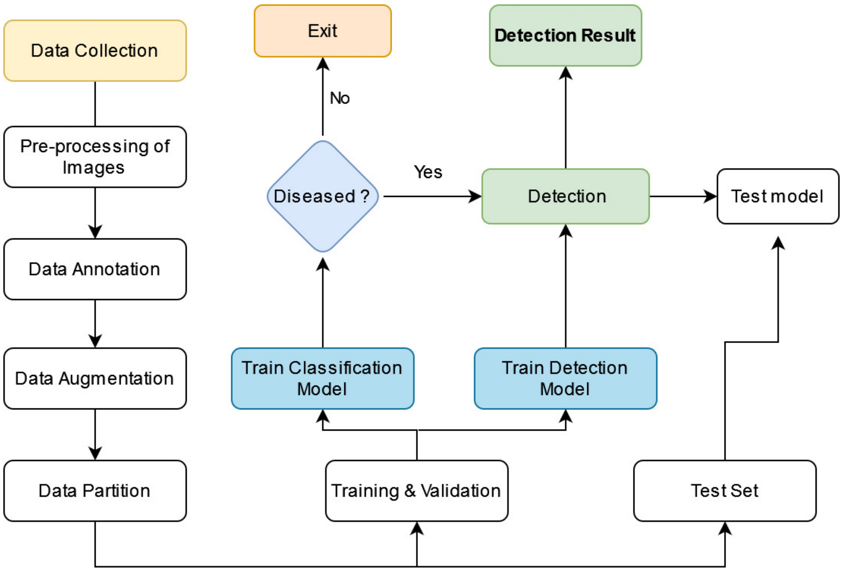


Figure 3: Pipeline of overall methodology

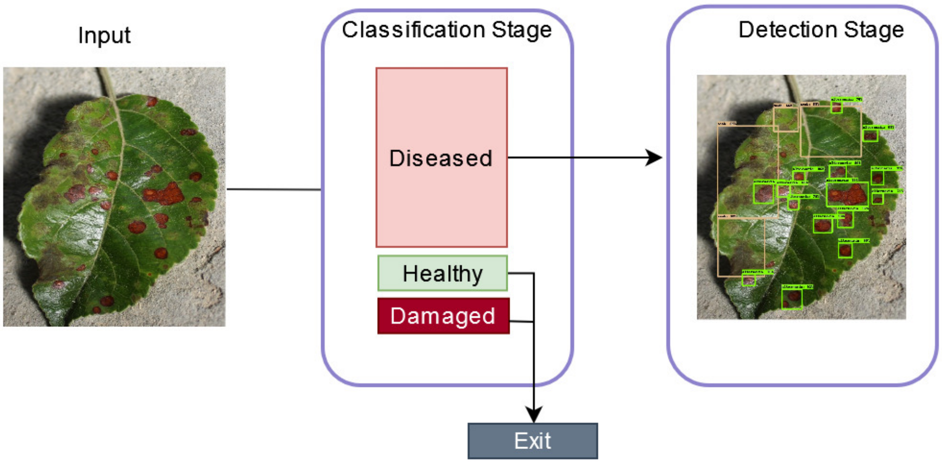


Figure 4: Overall structure of our detection system

ImageNet dataset. Xception uses depthwise separable convolution layers with residual connections instead of classical convolutions. Depthwise Separable Convolution replaces classic *n x n x k* convolution operation with *1 x 1 x k* point-wise convolution operation followed by channel-wise *n x n* spatial convolution operation. This way the number of operations are reduced by a factor proportional to 1/k. Residual connections are 'skip connections' which allow gradients to flow through a network directly, without passing through non-linear activation functions and thus avoiding the problem of vanishing gradients. In residual connections, output of a weight layer series is added to the original input and then passed through non-linear activation function as shown in Figure 5. We used Xception as the base model and added a Global Max-pooling layer followed by dropout layer and final fully-connected layer at the end. The resultant model has 20,881,970 parameters in total out of which 20,827,442 trainable and 54528 are non-trainable parameters. Architecture details, layer-wise parameters and output shape of the classification model are shown in Table II.

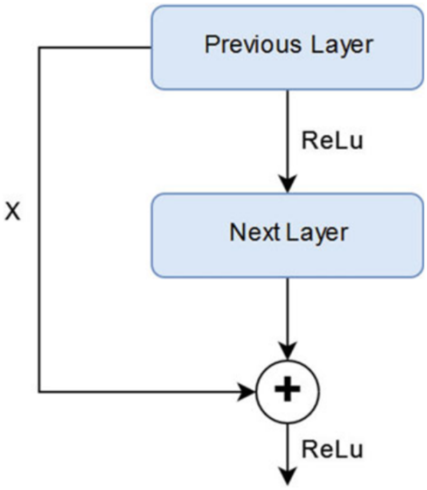


Figure 5: Residual Connection

Table II: Details of CNN Architecture

Layer (type)	Output Shape	Param #
Xception (Model)	5 x 5 x 2048	20861480
GlobalMaxPooling2D	2048	0
dropout (Dropout)	2048	0
dense (Dense)	10	20490

Total Parameters: 20,881,970

Trainable Parameters: 20,827,442

Non-trainable Parameters: 54,528

3.2. The Detection Stage

Object detection is the task of classifying and localizing objects in an image. In past recent years, the rapid advances of deep learning techniques have greatly accelerated the momentum of object detection. With advanced deep learning models and increasing computing power, the performance of object detectors has greatly improved, achieving many breakthroughs. Object detection approaches are broadly classified into two types, region proposal based approaches also known as two-stage approaches and single-stage approaches based on regression or classification (Nguyen et al., 2020). Single-stage object detection approaches are generally faster and hence preferred by many people.

Although object detection has come a long way in past one decade or so, it still has many key challenges to overcome. Some of these challenges include high intra-class variation, low inter-class variation and model efficiency. To overcome these challenges, researchers have come up with innovations like novel region proposals, multi-scale feature maps, divided grid cell, new loss function etc. These innovations have resulted in performance increase of object detection algorithms. However, one challenge which has attracted the attention recently is detection of small objects. Most of the state-of-the-art object detectors, both single-stage and two-stage approaches, have struggled with detecting small objects.

Our problem of disease detection is an object detection problem in which different symptoms have to be identified and localized. These symptoms can be of different sizes varying from few millimetres to few centimetres, thus making their detection difficult for most of the state-of-the-art detectors. A research conducted by Nguyen (Nguyen et al., 2020) who ran different models with different backbones on multi-scale objects to find out which model and backbone network is suitable for small objects. It has been observed that Faster R-CNN with ResNet-101+FPN backbone achieved the top mAP (mean Average Precision) of 41.2% on small object dataset. While as in single-stage approaches, Yolov3 608 × 608 with Darknet-53 obtained 33.1% mAP. As per the results above, Faster RCNN seems to be the ideal choice for our problem but since Yolov4 (Bochkovskiy et al., 2020) has been released which has improved the accuracy over Yolov3 by around 10% and also decreased the inference time by a margin, we decided to try both the approaches. Besides, we also choose another famous detection model EffecientDet (Tan et al., 2020) which has achieved better results than Yolo models on MS COCO dataset. All the three algorithms are best detection techniques among the lot.

a) YOLO v4

You Only Look Once (YOLO) is a family of one-stage object detection techniques first introduced in 2016. Yolov4 is the recent member of the family which is an enhanced version of its predecessor Yolov3. YOLO models are highly performant and efficient achieving state-of-the-art results on different applications of object detection. Yolo v4 is presently one of the best real-time object detection algorithms both in terms of Mean Average Precision (MAP) and speed. YOLO v4 combined many universal features (features which are independent of the dataset, models, and type of task) such as Weighted-Residual-Connections (WRC), Self-adversarial-training (SAT), Cross mini-Batch Normalization (CmBN), Cross-Stage-Partial-connections (CSP) and Mish-activation function, Mosaic data augmentation, DropBlock regularization, and CIoU loss (Bochkovskiy et al., 2020).

Yolov4 uses CSPDarknet53 as the backbone model and introduced some novel ways to improve the performance. Yolo v4 introduced new methods of data-augmentation Mossaic and Self-Adversarial Training (SAT). Mossaic combines 4 training images into one so that on each layer batch-normalization calculates activations from 4 different images instead of one. It also helps in keeping the batch size small with the same number of images. Self-Adversarial Training is another augmentation method in which an input image is first altered to create deception and then in the next stage the network is trained to detect an object from this altered image. In this way, Yolo v4 model executes an adversarial attack on itself. Yolo v4 also introduced CIoU loss for the bounding box which takes into consideration the distance between centre points, aspect ratio, and the overlapping area of the boxes. CIoU leads to faster convergence and better performance.

b) Faster RCNN

Faster R-CNN (Ren et al., 2015) is a region-based object detector which is an improved version of its predecessor Fast-RCNN (Girshick R, 2015). It first runs an image through a backbone network (CNN) and then on the last feature map of Convolution layers a fully convoluted network called region proposal network (RPN) is trained which outputs a set of bounding boxes along with their objectless scores which determine likelihood of an object. In our experiment, we used a ResNet101+FPN based Faster RCNN. ResNet101 is 101 layers deep Convolutional Neural Network. It obtains the best speed/accuracy trade-off. Feature Pyramid Network (FPN) (Lin et al., 2017) generates pyramid of feature maps at multiple scales useful for detecting small objects.

c) EfficientDet

EfficientDet is another state-of-the-art detection model by Google Brain Team which has achieved slightly better results on MS COCO dataset but at the cost of speed. EfficientDet uses weighted bi-directional feature pyramid network (BiFPN) for easy and fast multi-scale feature fusion and a single compound scaling factor that uniformly governs the width, depth and resolution for all backbone, feature network and prediction networks at the same time (Tan et al., 2020). EfficientDet uses EfficientNet as its backbone network which is a state-of-the-art CNN available developed by the same Google Brain Team. EfficientNet is small, efficient making it ideal for detectors.

3.3.Implementation and Training

Table III: Hyperparameters used for Classification Model

Hyper Parameter	Value
Weights	Imagenet
Learning Rate	1e-4
Epochs	100
Batch Size	16
Decay Rate	0.5
Patience	5
Dropouts	0.4
Activations	ReLu for Conv and Softmax for Classification
Loss Function	Categorical CrossEntropy

Table IV: # of instances of each disease in training set

Disease/Symptoms	# of Instances
Scab	5910
Alternaria	7674
Mossaic	1716
MLB only	1180
Powdery Mildew	1795
Necrosis	2490
Insect	881

In this section, we present the details of our experimental setup, training and datasets used for evaluation.

Classification Model (Stage 1): The classification model Xception was implemented in Keras on top of Tensorflow on a workstation with 16GB RAM, Intel (R) Xeon (R) @ 2.30Ghz processor with Tesla K80 12GB graphics card. Model parameters were initialized through the process of Transfer learning and then re-trained on prepared dataset of around 9000 images (after augmentation) using stochastic gradient descent (SGD) with learning rate of 0.0001, input size 460, batch size of 16 and epoch value of 100. The model was further fine-tuned using Adam optimizer with a learning rate of 0.00005. All the hyper-parameters and their values are given in Table III below.

Detection Stage (Stage 2): To train the detection models, the dataset was split into 70% train, 20% validation and 10% test sets. The models were trained on training set, and finally tested on the test set. The number of total instances of each class in our training set is given in Table IV. Our first model that is Yolov4 was implemented using Darknet Framework. Darknet is a custom open source neural network framework written in C and CUDA by Joseph Redmon (Redmon Joseph, 2013). As a first step, the annotations done using LabelImg (xml files) have to be converted in Darknet format. In this format, a text file is generated for each image which contains the annotations and a numeric IDs. A labelmap file is also generated which maps the ID's to string labels (class names). We used Roboflow (Roboflow, 2020) to convert the dataset into required format. Roboflow provides online

Table V: Detection models and their hyper-parameter values

Method	Momentum	Decay	Gamma	Lr	Batch Size	Training Time	Stepsize	Input Size
EfficientDet-B0	0.89	0.0005	0.1	0.001	16	12hrs	11000	600
Yolov4	0.843	0.00036		0.003	8	8hrs	2000	416
Faster R-CNN	0.9	0.0001	0.1	0.001	16	6hrs	5000	600

services to annotate and convert annotations to or from any other format. We trained Yolov4 for 2000 epochs with input size of 416x416 after which the model performance stopped showing any improvement.

Our second model Faster-RCNN was implemented in Detectron2. Detectron2 is Facebook AI Research's (FAIR) object detection library written in PyTorch. Detectron2 is flexible and extensible, allowing high-quality implementations of state-of-the-art object detection algorithms. We used Detectron2's implementation of Faster-RCNN and trained the model for 5000 iterations with the initial learning rate of 0.01 for first 1000 iterations and then 0.001 for next 4000 iterations. The model was trained using stochastic gradient descent (SGD) optimization method, the gamma value was 0.1, momentum 0.9, and weight decay of 0.0001.

Our final model EfficientDet-D0 was implemented in Keras and Tensorflow2 and trained for 11000 steps with batch size 16, input size 600, and gamma value 0.1. The optimization method used was momentum optimizer with momentum optimizer value of 0.899. All the hyper-parameters and their values for each model are given in Table V below.

4. RESULTS AND DISCUSSIONS

4.1. Quantitative Results

This section presents performance evaluation of both classification and Detection models.

Performance of Classification model

We evaluated the proposed classification model on hold-out test set. The multi-class classification results were recorded and then average numbers were calculated. The performance of the model is presented in the form of Confusion matrix (CM) in Figure 6. Overall Accuracy, precision, recall and F-measure computed by formulae given below are summarized in Table VI.

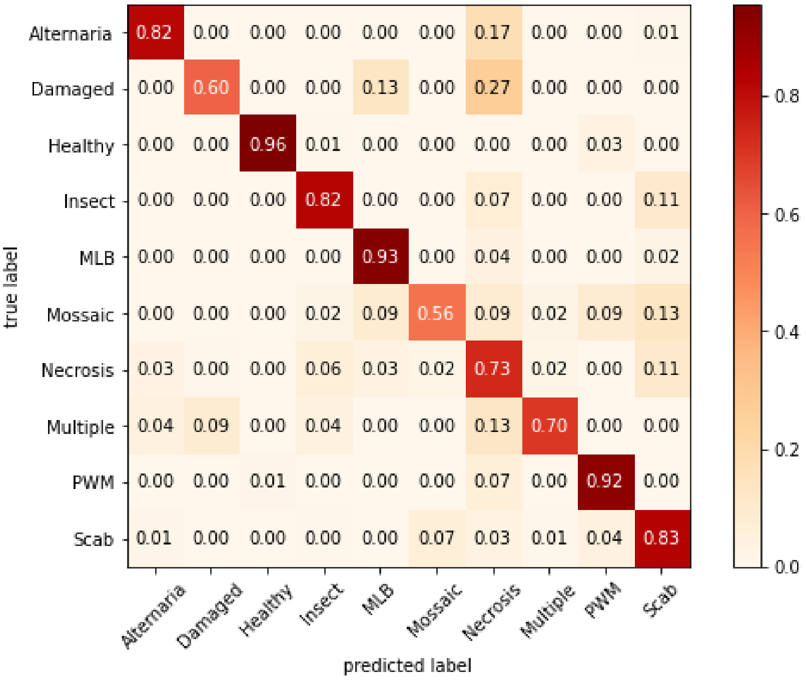
Accuracy = $\frac{No.\,of\ images\ correctly\ classified}{Total\,no.\,of\ images}$

Precision = $\frac{Sum\ of\ all\,True\ Positives\ (TP)}{Sum\ of\ all\,True\ Positives\ (TP) + All\ False\ Positives\ (FP)}$

Recall = $\frac{Sum\ of\ all\,True\ Positives\ (TP)}{Sum\ of\ all\,True\ Positives\ (TP) + All\ False\ Negatives\ (FN)}$

F-measure = $\frac{2 * Precision * Recall}{Precision + Recall}$

The aforementioned performance metrics are the top metrics used to measure the performance of classification models. The classification model achieved an average accuracy of 81.09%, precision



	Alternaria	Damaged	Healthy	Insect	MLB	Mossaic	Necrosis	Multiple	PWM	Scab
Alternaria	0.82	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.00	0.01
Damaged	0.00	0.60	0.00	0.00	0.13	0.00	0.27	0.00	0.00	0.00
Healthy	0.00	0.00	0.96	0.01	0.00	0.00	0.00	0.00	0.03	0.00
Insect	0.00	0.00	0.00	0.82	0.00	0.00	0.07	0.00	0.00	0.11
MLB	0.00	0.00	0.00	0.00	0.93	0.00	0.04	0.00	0.00	0.02
Mossaic	0.00	0.00	0.00	0.02	0.09	0.56	0.09	0.02	0.09	0.13
Necrosis	0.03	0.00	0.00	0.06	0.03	0.02	0.73	0.02	0.00	0.11
Multiple	0.04	0.09	0.00	0.04	0.00	0.00	0.13	0.70	0.00	0.00
PWM	0.00	0.00	0.01	0.00	0.00	0.00	0.07	0.00	0.92	0.00
Scab	0.01	0.00	0.00	0.00	0.00	0.07	0.03	0.01	0.04	0.83

Figure 6: Confusion matrix of our classification results

Table VI: Class-wise results of classification model

Disease	**Precision**	**Recall**	**F1-Score**	**Actual/ Classified**
Alternaria	0.90	0.82	0.86	116/105
Damaged	0.86	0.60	0.71	30/21
Healthy	0.99	0.96	0.97	90/87
Insect	0.56	0.82	0.67	28/41
MLB	0.85	0.93	0.89	91/100
Mossiac	0.58	0.56	0.57	45/43
Multiple	0.76	0.73	0.75	226/219
Necrosis	0.70	0.70	0.70	23/23
PWM	0.81	0.92	0.86	75/85
Scab	0.83	0.83	0.83	212/212
Average	**0.784**	**0.787**	**0.781**	
Overall Accuracy	**81.09%**			

and recall of 78.4% and 78.7% respectively. The performance of a classifier can reduce significantly when faced with multiple classes with minimum variation. Leaf images infected with different diseases and with different intensities can confuse the classification model and thus result in lower performance. For example, in the confusion matrix above, it can be seen that many Alternaria examples have been classified as Necrosis Blotch. This can happen if some alternaria spots turn tan or black papery spots with time, making it appear as some other symptom like Necrosis etc. Similarly, compared to other classes, the model is more prone to confusion in distinguishing between Multiple diseases and Damaged. This is due to the fact that multiple overlapping spots damage the leaf tissues.

If we combine all the single diseases into one single class as Diseased class, then the overall accuracy increases significantly. We did slight modification to the same classification model and fine-tuned it for four classes only. After fine-tuning, the model was tested on test set comprising of 340, 226, 90 and 30 images belonging to 'Diseased', 'Multiple', 'Healthy' and 'Damaged' classes respectively. The Confusion matrix is presented in Figure 7. It can be seen that the overall accuracy of the model increased from 81.09% to 87.9%. In addition, class-wise precision, recall, F-measure, and accuracy are given in Table VII.

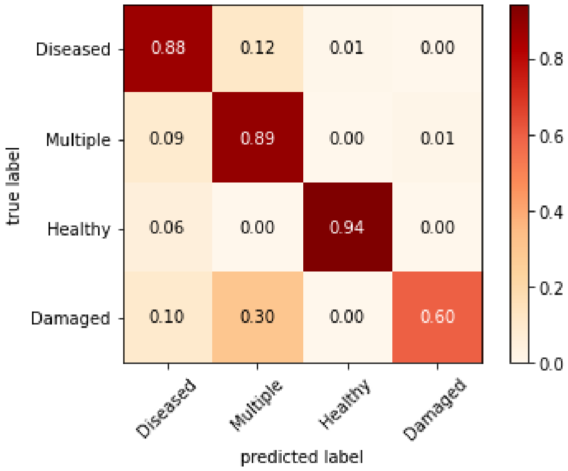


Figure 7: Confusion matrix of our 4-class classification results

Table VII: Class-wise results of 4-class classification model

Disease	Precision	Recall	F1-Score	Actual/ Classified
Diseased	0.91	0.88	0.89	340/327
Multiple	0.80	0.89	0.85	226/251
Healthy	0.98	0.94	0.96	90/87
Damaged	0.86	0.60	0.71	30/21
Average	**0.88**	**0.82**	**0.85**	
Overall Accuracy	**87.9%**			

Performance of Detection models

We evaluated the performance of detection models on a test set from our prepared Apple Disease dataset. The test set comprises of around 10% of the total images. We used mean Average Precision (mAP) metric to evaluate the performances. Mean AP is the main evaluation indicator used to evaluate detection models. The test results of all the detection models is presented in Table VIII. Out of the three detection models, Faster RCNN with mAP of 42.1% outperformed EfficientDet and Yolov4. After Faster-RCNN, next best model was Yolov4 with mAP of 41.4% and EfficientDet with mAP of 38% came third. According to the results, all the three models achieved best mAP score for Powdery mildew (PWM) category while as scab is the worst performing category. Scab disease spots vary immensely in shape and size. They can be black or olive-green or sometimes purple in colour. As the number of spots grow they can turn sooty. On the other hand PWM appears as white powder on leaves which very often covers whole leaf surface thus easily identifiable.

Table VIII: Class-wise Test results (AP) of our three detection models

Model **Disease**	**EfficienDet**	**Yolov4**	**Faster-RCNN**
Alternaria	39.75	53.75	**58.1**
Insect	42.6	55.6	**57.6**
MLB	19.4	28.4	**30.4**
Mossiac	26.3	**32.3**	31.3
Necrosis	23.8	30.8	**32.1**
PWM	56.1	**58.4**	**59.2**
Scab	16.9	**27.9**	25.9
mAP	32.1	41.1	**42.01**

4.2. Qualitative Results

We evaluated the performance of our best detection model for all categories in our Apple Disease dataset. Our detector is effectively able to detect various complex symptoms easily. It shows excellent results due to its robustness to deal with objects of different shapes, scales, colours, etc. Figure 8 shows some example results predicted by our proposed detector on different types of diseases.

From qualitative results, it is evident that our model has been very accurate in detecting various diseases. However, the quantitative results are not very ideal. Faster-RCNN has performed slightly better than the rest of the two models, but it has achieved low mAP and we can see a lack of performance in some categories. This is due to the fact that evaluating object detection models is not as straightforward as evaluating classification models. The reason is that an image can contain

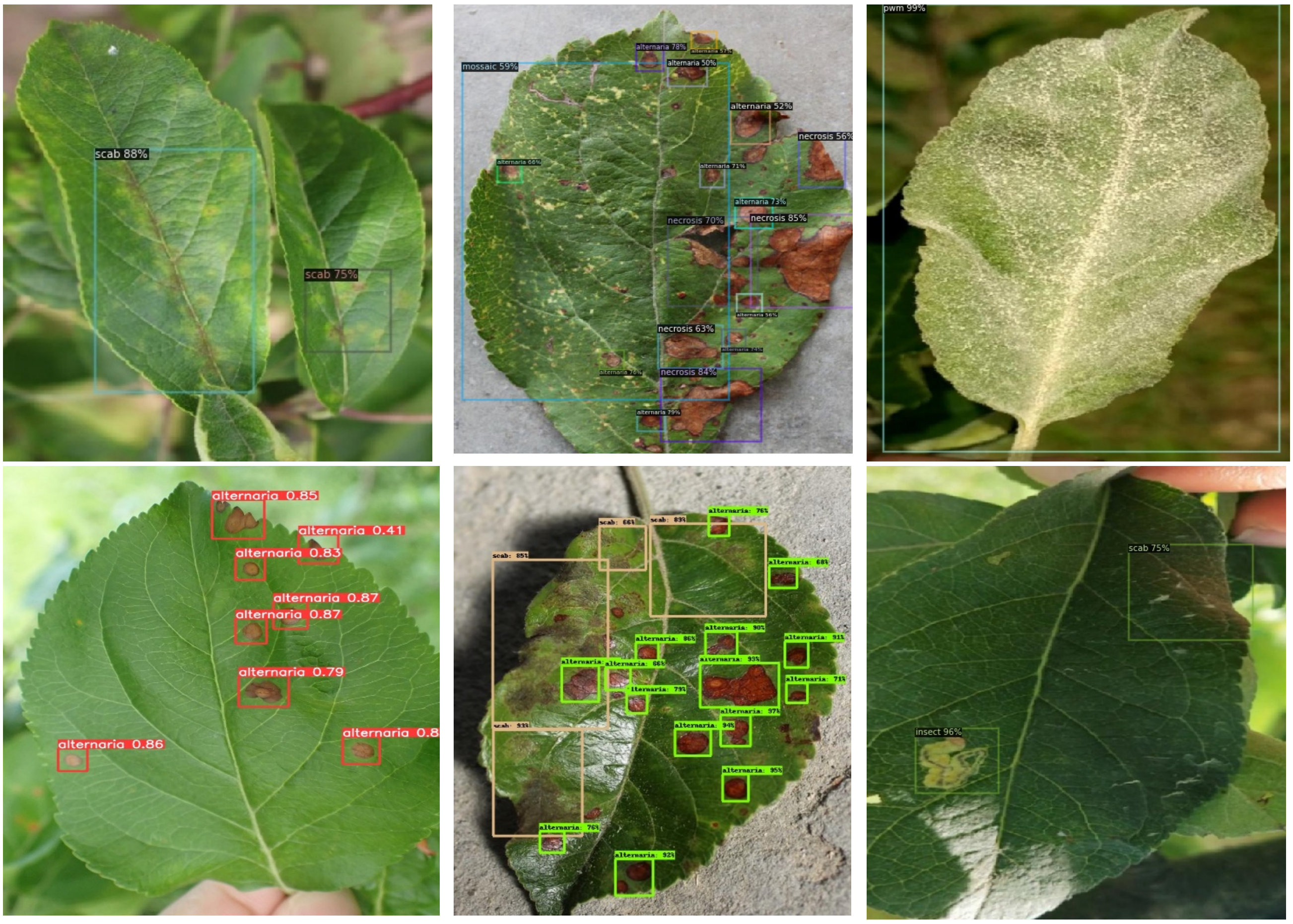


Figure 8 : Results predicted by our detection models

1) Scab 2) Multiple diseases (Mossaic, Alternaria, Necrosis) 3) Powdery Mildew 4) Alternaria 5) Alternaria and Scab 6) Insect and Scab

many objects belonging to different classes. So, in order to evaluate an object detection model, three things have to be verified; object class, corresponding location (bounding box) and confidence. First find all the object in an image and then verify if all the found objects belong to their correct classes. In a detection task, where different objects overlap and come in different shapes and sizes, mAP can be very misleading as we have observed in our experiments.

Undoubtedly, mAP successfully summarizes the performance of an object detection model in one number but unravelling different errors from mAP is challenging because all forms of errors like misclassification, miss-localization or duplicate detection all contribute to false positives. All these errors are not same but have equal effect on mAP. For example, misclassification means your model is not generalizing well and is a serious performance issue while as duplicate detection means the model is able to detect objects but redundantly. In tasks where number of detected objects is not important, duplicate detections can be ignored.

In order to understand the quantitative performance of our model we analysed the results from our model and found out many duplicate detections especially for Scab disease. It is the same category for which the mAP score is low as compared to other categories. Figure 9 shows some of the examples of duplicate detections.

Another challenge that we were dealing with was detecting small spots. Our all three detection models performed well in detecting different spots of moderate to large sizes, however Faster-RCNN outperformed the other two models in detecting small spots as well. Figure 10 presents the comparison of all three models in detecting small spots from leaf images. EfficientDet performed well on medium and large sized spots but missed most of the small spots as shown in Figure 10. The performance of Yolov4 on small size spots was better than EfficientDet but not as good as Faster-RCNN.

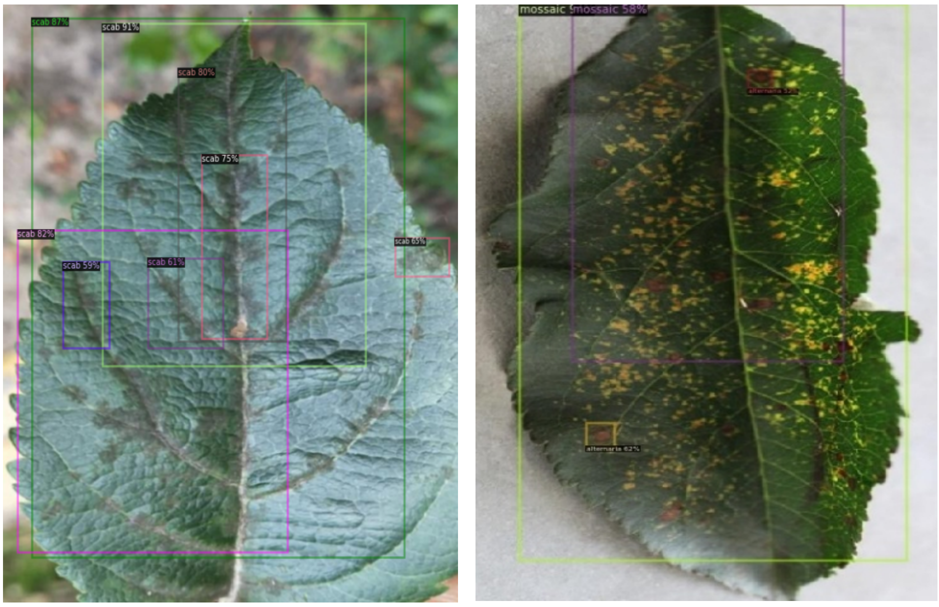
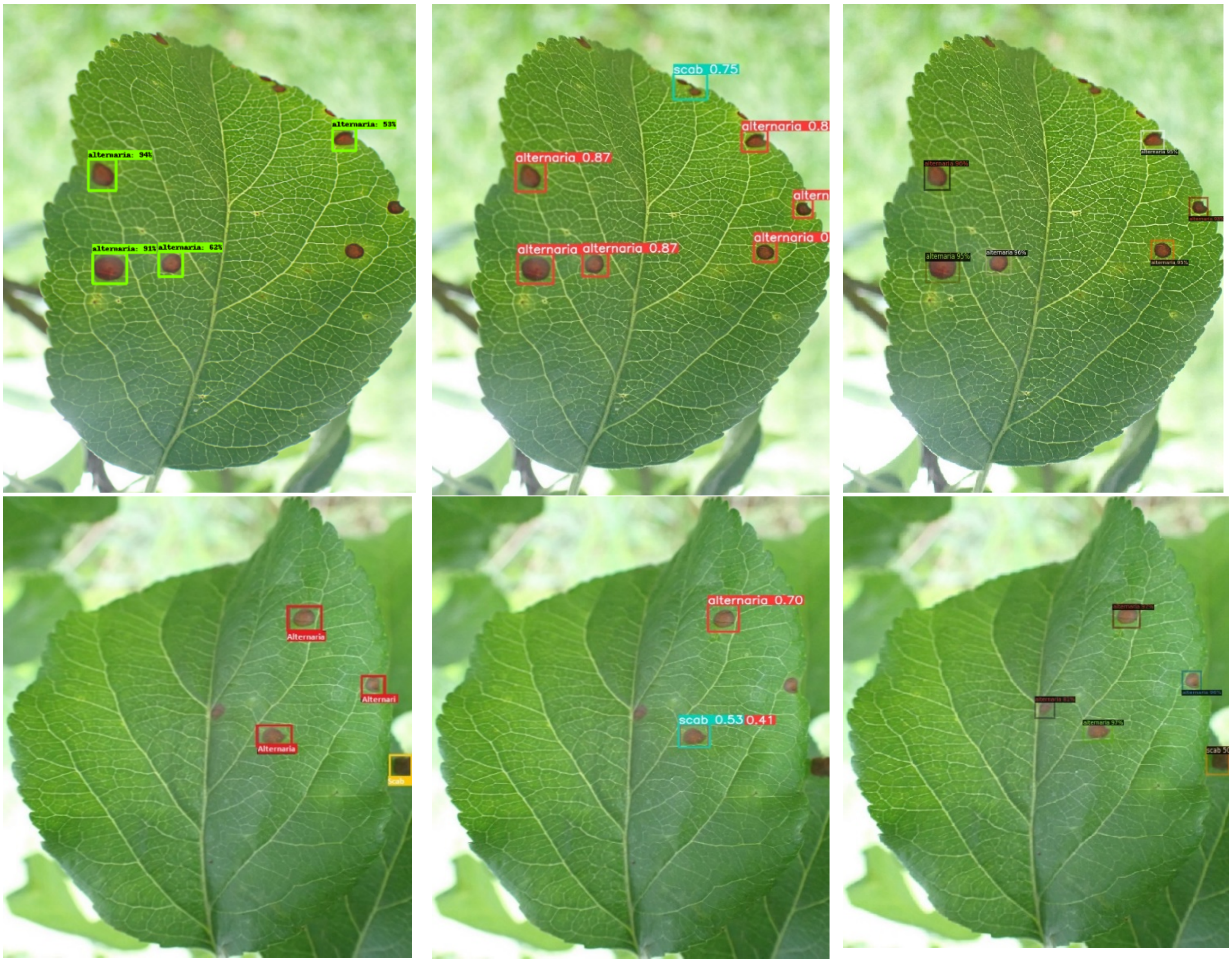


Figure 9: Duplicate (overlapped) Detections
a) 6 duplicate scab detections b) 1 mossaic duplicate detection

5. CONCLUSION

Plant disease detection is a challenging problem due to variable disease patterns, complex background and lighting conditions. In this research work, we constructed an expert-annotated apple disease dataset of suitable size consisting of around 9000 high quality RGB real-field images covering all the main foliar diseases and symptoms. Next, we proposed a deep learning based apple disease detection system which can efficiently and accurately identify the symptoms. The proposed system works in two stages, first stage is a tailor-made light weight classification model which classifies the input images into diseased, healthy or damaged categories and the second stage (detection stage) processing starts only if any disease is detected in first stage. Detection stage performs the actual detection and localization of each symptom from diseased leaf images. Both classification and detection models were initialized using transfer learning and then trained on prepared dataset. Data augmentation techniques like rotation, translation, noise, cut-out etc. were also applied to prevent overfitting. The results demonstrate that the proposed detection system can detect various types of apple leaf diseases with high accuracy in real time. It can be very helpful tool for farmers and apple



EfficientDet Yolov4 Faster-RCNN

Figure 10: Model performance on small size spots

growers to aid them in diagnosis, quantification and follow-up of infections. Furthermore, in future, the work can be extended to other fruits and vegetables as well.

Acknowledgements

The study is part of the project "*Deep Learning for Apple Diseases: Development of an Intelligent System for Detection of Apple Fruit Diseases*" sponsored by the UGC Dr. D. S. Kothari Post-Doctoral Fellowship Scheme

Declaration of Competing Interest

The authors have no conflict of interest to disclose.

References

Abbas A, Jain S, Gour M, Vankudothu S. Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture. 2021 Aug 1;187:106279.

Amara J, Bouaziz B, Algergawy A. A deep learning-based approach for banana leaf diseases classification. Datenbanksysteme für Business, Technologie und Web (BTW 2017)-Workshopband. 2017.

Bhat, K. A., Peerzada, S. H., & Anwar, A. (2015). Alternaria epidemic of apple in Kashmir. African Journal of Microbiology Research, 9(12), 831-837.

Bochkovskiy A, Wang CY, Liao HY. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. 2020 Apr 23.

Brahimi, Boukhalfa, Mohammed, Kamel & Moussaoui, Abdelouahab, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization". 2017.

Chollet F. Xception: Deep learning with depthwise separable convolutions. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 1251-1258).

Chuanlei Z, Shanwen Z, Jucheng Y, Yancui S, Jia C. Apple leaf disease identification using genetic algorithm and correlation based feature selection method. International Journal of Agricultural and Biological Engineering. 2017 Mar 31;10(2):74-83.

Directorate of Horticulture, Kashmir. Available from: http://hortikashmir.gov.in/Area Production data.html [Accessed 10-Nov-2021]

Dubey SR, Jalal AS. Detection and classification of apple fruit diseases using complete local binary patterns. In2012 Third International Conference on Computer and Communication Technology 2012 Nov 23 (pp. 346-351). IEEE.

Dutot M, Nelson LM, Tyson RC. Predicting the spread of postharvest disease in stored fruit, with application to apples. Postharvest biology and technology. 2013 Nov 1;85:45-56.

Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture. 2018 Feb 1;145:311-8.

Girshick R. Fast r-cnn. InProceedings of the IEEE international conference on computer vision 2015 (pp. 1440-1448).

Hu, G., Wu, H., Zhang, Y., Wan, M., 2019. A low shot learning method for tea leaf's disease identification. Comput. Electron. Agric. 163, 104852. https://doi.org/10.1016/j.compag.2019.104852. ISSN 0168-1699

Jiang P, Chen Y, Liu B, He D, Liang C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access. 2019 May 6;7:59069-80.

Kawasaki Y, Uga H, Kagiwada S, Iyatomi H. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. InInternational symposium on visual computing 2015 Dec 14 (pp. 638-645). Springer, Cham.

Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 2117-2125).

Liu B, Zhang Y, He D, Li Y. Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry. 2018 Jan;10(1):11.

Nguyen ND, Do T, Ngo TD, Le DD. An evaluation of deep learning methods for small object detection. Journal of Electrical and Computer Engineering. 2020 Apr 27.

Prasanna Mohanty S, Hughes D, Salathe M. Using Deep Learning for Image-Based Plant Disease Detection. arXiv e-prints. 2016 Apr:arXiv-1604.

Raashid Hassan. Grim omen for apple growers as scab noticed on trees after plenty of rain. Available from: https://kashmirreader.com/2021/04/19/grim-omen-for-apple-growers-as-scab-noticed-on-trees-after-plenty-of-rain 19th April 2021 [Accessed 10-Nov-2021]

Ramcharan A, Baranowski K, McCloskey P, Ahmed B, Legg J, Hughes DP. Deep learning for image-based cassava disease detection. Frontiers in plant science. 2017 Oct 27;8:1852.

Redmon Joseph, Darknet: Open Source Neural Networks in C, 2013, http://pjreddie.com/darknet

Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;28:91-9.

Roboflow, 2020, https://roboflow.com [Accessed 15-Nov-2021]

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 2818-2826).

Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object detection. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020 (pp. 10781-10790).

Tzutalin L. Git code (2015). Availabel from: https://github.com/tzutalin/labelImg [Accessed 2020 Apr].

Velasquez AC, Castroverde CD, He SY. Plant–pathogen warfare under changing climate conditions. Current Biology. 2018 May 21;28(10):R619-34.

Zhang, Zhang, Q., Li, P., 2019. Apple disease recognition based on improved deep convolution neural network. J. Forest. Eng. 4 (04), 107–112.

Zhong Y, Zhao M. Research on deep learning in apple leaf disease recognition. Computers and Electronics in Agriculture. 2020 Jan 1;168:105146.