Plant Disease Detection Using Sequential Convolutional Neural Network

Anshul Tripathi[1*], Uday Chourasia[1], Priyanka Dixit[1], and Victor Chang[2*]

1 [1] Department Of Computer Science And Engineering, University Institute of Technology, RGPV, Bhopal (MP), India; anshultripathi731@gmail.com

2 2. Artificial Intelligence and Information Systems Research Group, School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, UK 2; ic.victor.chang@gmail.com and V.Chang@tees.ac.uk

* Correspondence: e-mail [1*] anshultripathi731@gmail.com and [2*] ic.victor.chang@gmail.com (optional; include country code; if there are multiple corresponding authors, add author initials)

Abstract: Plant disease cause great damage in agriculture. In this paper, simple leaves images of healthy and diseased plants were used to detect and diagnose plant diseases through CNN Model. All important steps required for implementing these disease recognition models are completely described throughout the paper, starting from gathering images to make the database evaluated by agricultural experts a deep learning framework to perform the deep CNN training. The main warning in the area of food preservation and care is on topmost are crop diseases. It has been recognized speedily, but it is not as easy as in any area of the world because no required framework exists. Both the healthy and diseased plant leaves were gathered and collected under the condition and circumstances. For this purpose, a public set of information was used. It was 20,639 images of plants that were infected and healthy. In order to recognize three different crops and 12 diseases, a sequential convolutional neural network from Keras was trained and applied. The perfection and accuracy of the model is 98.18 % onset of information of the above trained mentioned model using CNN. It has also indicated the probability and possibility of this strategy and procedure. The over-fitting occurs and neutralizes by putting the dropout value to 0.25.

Keywords: Plant Disease; Convolution Neural Network; Extractness; Infected Leaves; Healthy Leaves

1. Introduction

Food and Agriculture Organization of the United Nations Organization has supposed that food production increase by 70% until 2050 to fulfill the necessity of food in the world. Modern technology nowadays has provided ample space to produce to meet the demand of more than 7 billion people. Agriculture is the foremost occupation in India. In the whole world, India has secured its second rank in agriculture. There is a high chance of food insecurity if production is not increased. It can only be enhanced or increase using the exact topmost technology. Several factors are responsible for food security threatening. Among these is crop disease is the foremost threat to food security (Dhakal & Shakya, 2018). In earlier times, trained experts were used to detect diseases in the plant through the naked eye or visual inspection. It was costly and inappropriate, as there was a lack of human intelligence also. Machine Learning was used to solve this problem in which CNN is its part of detecting the disease by preprocessing infected plants leaf and fitting into a neural network. According to Karol et al. (2019), by using image preprocessing, the first implementation of plant disease detection was performed by SHEN WEIGHEH WUYACHUN CHEN ZHANLIANG and WI HANGDA in their paper. According to (Wallelign et al.,2018) they state that machine learning methods such as Artificial Neural Network (ANN) Decision Trees, K-means, K nearest neighbors, and Support Vector Machines (SVM) are used and applied in agriculture research. Like Machine Learning, Deep learning also contains supervised, unsupervised, semi-supervised, and reinforcement learning methods or models. Among the supervised models are applied in agriculture to perform different tasks such as image segmentation, image classification, and object detection. The production quality and quantity of tomato crops are affected by the wide-scale presence of disease. A study was completed by (Purushothaman et al., 2018) to counter the problem; images of tomato leaves were derived from plant village datasets as input to two deep learning-based architectures named Alex Net and VGG 16 Net. (Sumbwanyambe & Sibiya,2019) states that grey leaf spot by maize disease is caused by Cercospora maydis fungus. It is supposed a serious threat to maize production in large areas of the US corn belt and Africa. A smartphone camera model was developed to recognize three different maize leaf disease types out of the healthy leaf. The disease was northern corn leaf blight, common rust, and grey leaf spot. Common rust maize disease caused by the Puccinia sorghi pathogen is favored by coo temperature (16-23 degrees Celcius) and high radiative humanity (100%). Moreover, spots are found on both upper and lower leaf surfaces. The present model can identify the following diseases in Potato Crops, Tomato leaves, and Pepper Bells. These are Bacterial Spots in Pepper Bell. Early Blight and Late Blight in Potato. Bacterial Spot, Late Blight, Early Blight, Leaf Mold, Septoria Leaf Spot, Spider mites Two-spotted spider mite, Target Spot, Tomato Yellow leaf Curl virus, and Tomato Mosiac virus in Tomato Plants. So for detecting the plant disease, a CNN model is developed. The aim and goal of our research are to solve the problem of detection and prevent diseases of agricultural crops. India is a land where agriculture is the main occupation of people. Farmers are simple and unaware of technologies in this field. Various methods have been developed to diagnose disease. These methods are unavailable for many farmers and need domain knowledge or a great amount and resources to carry out. This study bridges a gap between farmers and technology by using CNN in detecting diseases in plants.

2. Materials and Methods

Dataset Description

To begin this research, data collection for high-quality data is required. The data was collected from Plant Village Dataset containing 20,639 images of

Diseased Plant Leaves. This dataset contains three different types of crops and 12 diseases. The size of the dataset is 326.38 MB.

In this dataset, there are 15 diseases of Potato, Tomato, and Pepper Bell. Table 1 below showing the details of the dataset.

Table 1

Sr. No.	Disease Name	Image	Number Of Images
1	PEPPER BELL BACTERIAL SPOT.	 Figure 1	997
2	PEPPER BELL HEALTHY	 Figure 2	1478
3	POTATO EARLY BLIGHT	 Figure 3	1000
4	POTATO LATE BLIGHT	 Figure. 4	1000

5	POTATO HEALTHY	 Figure.5	152
6	TOMATO BACTERIAL SPOT	 Figure.6	2127
7	TOMATO EARLY BLIGHT	 Figure.7	1000
8	TOMATO LATE BLIGHT	 Figure.8	1909
9	TOMATO LEAF MOLD	Figure.9 	952

10	TOMATO SEPTORIA LEAF SPOT	 Figure.10	1771
11	TOMATO SPIDER MITES TWO-SPOTTED SPIDER MITE	 Figure.11	1676
12	TOMATO TARGET SPOT	Figure.  12	1404
13	TOMATO YELLOW LEAF CURL VIRUS	 Figure.13	3209
14	TOMATO MOSIAC VIRUS	Figure.14 	373

15	TOMATO HEALTHY	 Figure.15	1591

This research can be conducted by either taking the benchmark dataset or the real-time dataset. Different types of sensors are required if analyzing real-time data. The sensors will be used to capture and developing the own dataset. This research can be expanded by applying different CNN layers and can be developed under an environment with a suitable architecture. This includes using Raspberry Pi to analyze the diseases of various plants in the home garden.

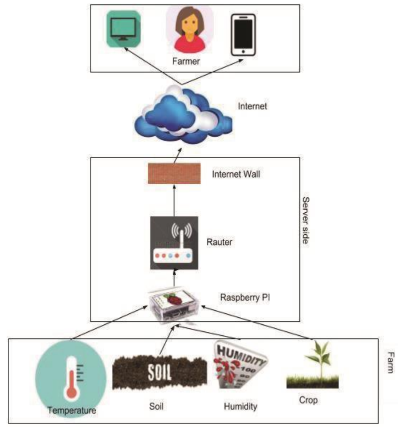


Figure 16. The architecture of Prototype

This architecture can be implemented in-house by deploying various smart sensors for measuring humidity, temperature, and soil moisture detection of disease will also be done by using a camera. Data collected will be sent to Raspberry Pi using wireless devices.

The data is verified and matched with the ideal data and soil moisture data values of humidity and temperature on the server-side. If there is any mismatch occurred, then the notification will be sent to the house owner.

Plant disease detection is done by applying CNN layers. The image is collected using a camera and sent to the server-side, where the CNN operation will do in the model.

Flow Chart

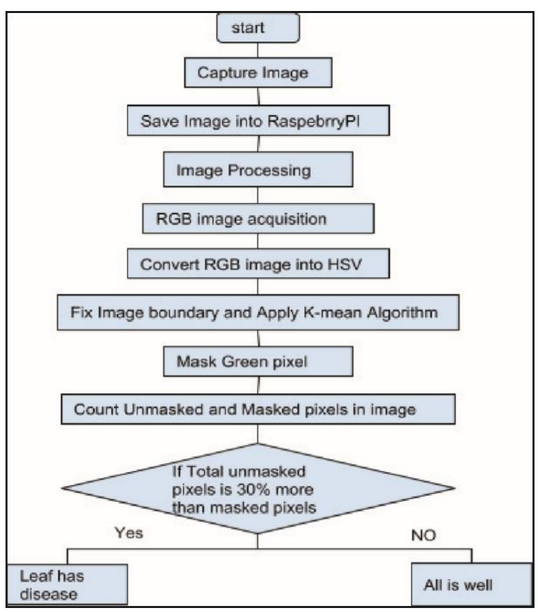


Figure 17. Showing steps to detect the condition of the leaf.

The workflow chart of Convolutional Neural Network is describing the step-by-step process in Figure 18. Initially, the data is collected after the Image Acquisition process takes place and after this process, image preprocessing and Image Segmentation are done after this Feature Extraction occurs. Training and Testing of the dataset are done after feature extraction. After this, classification is done by the classifier. At last, we get the output, which is bifurcated as Normal data or Diseased data.

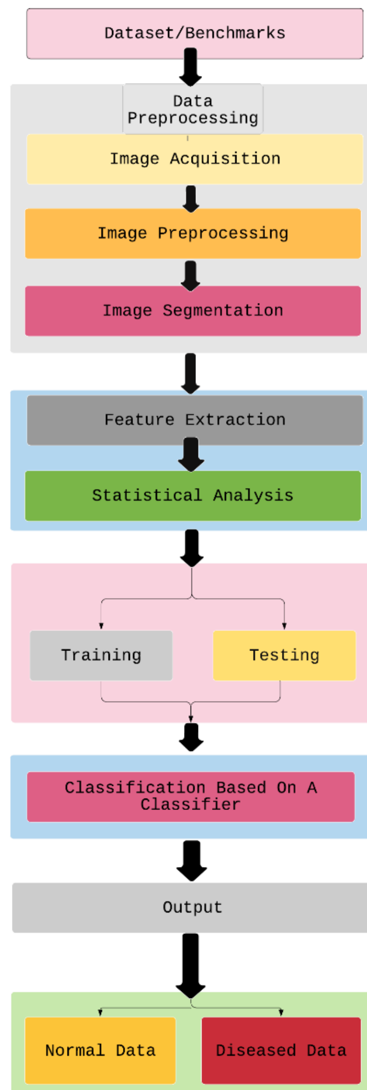


Figure 18. Work Flow

On below, the methodology is showing through a chart in Figure 19.

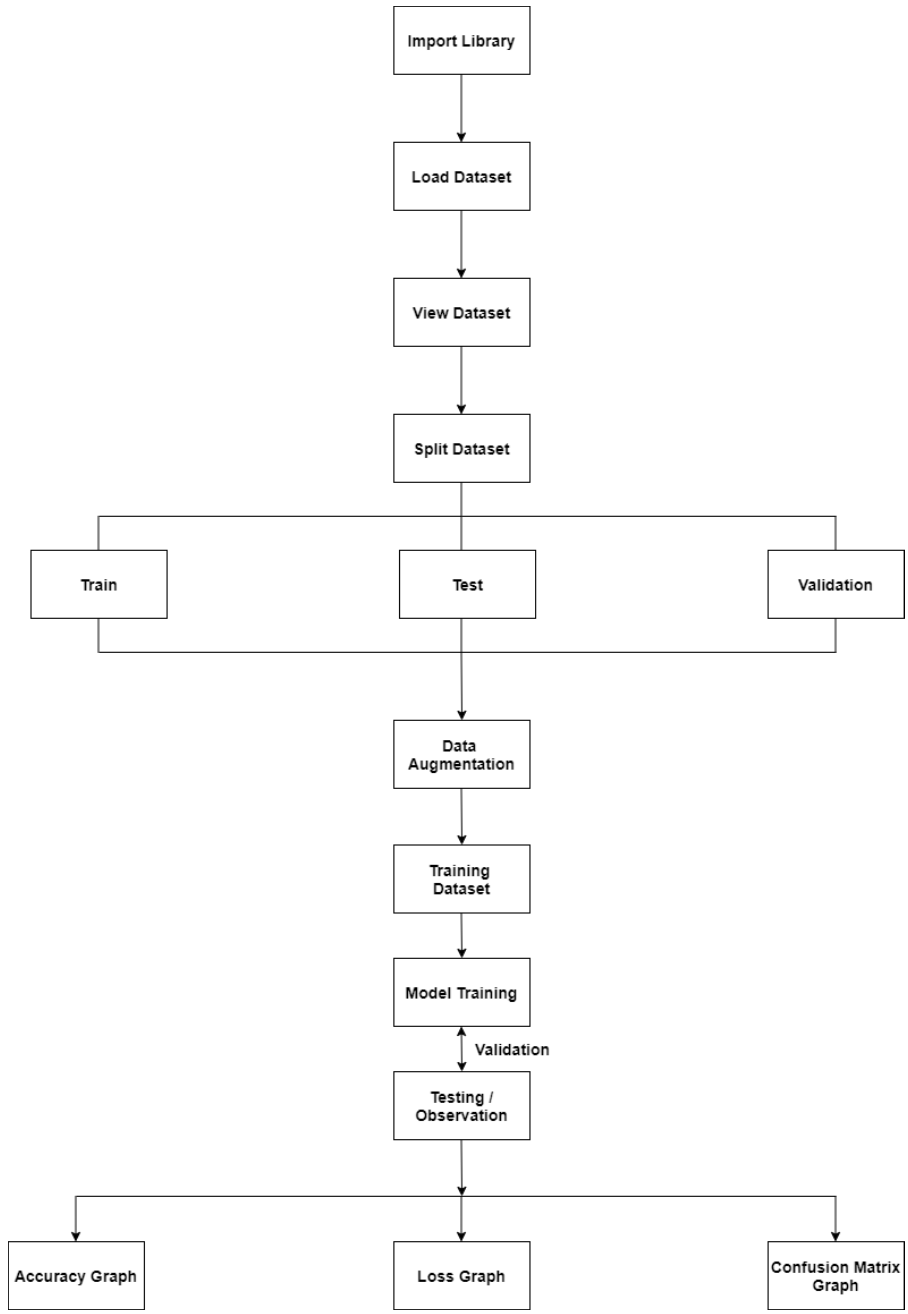


Figure.19. Methodology

The methodology consists of eight steps as below.

1). Import Library – The first and foremost thing I have done during the process that has imported the library and the necessary packages to further the execution model.

2). Load Dataset – Moving towards the next step, loaded the dataset on the model. The dataset, which was in the form of images of healthy and diseased leaves, has been loaded to the model, or the dataset can be loaded by taking real-time images.

3). View Dataset – While viewing the dataset, the data visualization was loaded earlier.

4). Split Dataset – In this step, the division of the whole dataset into two parts trained dataset and the test dataset. The above-mentioned trained dataset comprises 80% dataset, while on the other hand, the remaining 20% dataset contains the test dataset. After this, on following step validation process is carried out.

5). Data Augmentation – Now, we switch towards data augmentation. In this process, we reproduce new data that can be called artificial data from existing data. The following process is used in data augmentation, Scaling, Translation, Rotation of 90%, Rotation at finer angles, flipping, adding salt and pepper noise, perspective transform.

6). Training Dataset – The dataset we obtained after the data augment the process, we train that particular dataset for the advanced stage.

7). Model Training – The dataset which we obtained after the training dataset. We use the data to train the model. The validation process is also carried out in this model training to validate the data.

8). Testing and Observation – At this stage, the 20% test data kept separately is now applied and inserted in the model trained above. In the end, we get the result in the form of an Accuracy Graph, Loss Graph and Confusion Matrix, and ROC Curve.

LAYERS OF CONVOLUTIONAL NEURAL NETWORK

CONVOLUTION LAYER

For more filtering and clarifying an RGB image as an alternate layer, the layer collects input or output. In order to construct a feature map that exhibits low-level features like edges and curves, collected input is read as pixel values. We can recognize higher-level distinctive properties through a series of extra and fresh layers of convolution. Image is treated as a volume of size W x H x D. It displays a color image exemplifies in tensor flow and the matrix shows each channel, namely blue, green, and red.

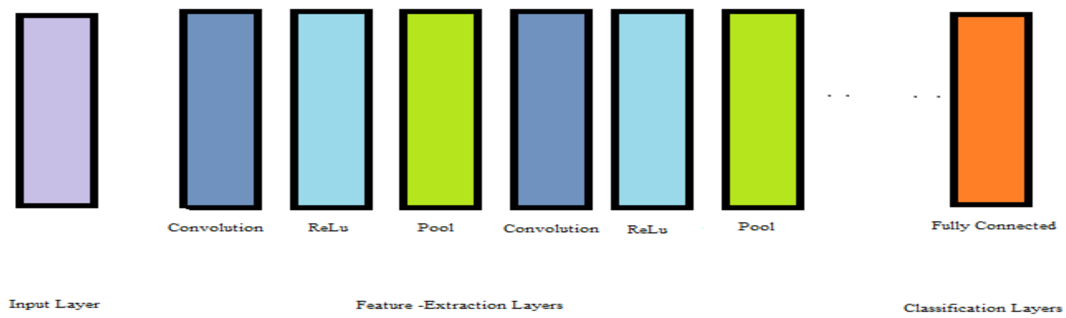


Figure.20 CNN ARCHITECTURE

BATCH NORMALIZATION LAYER

This layer is used to normalize the activations of prior layers at each and every batch. Transformation techniques apply to put the mean activation to nearly 0 but put the standard deviation nearly to 1.

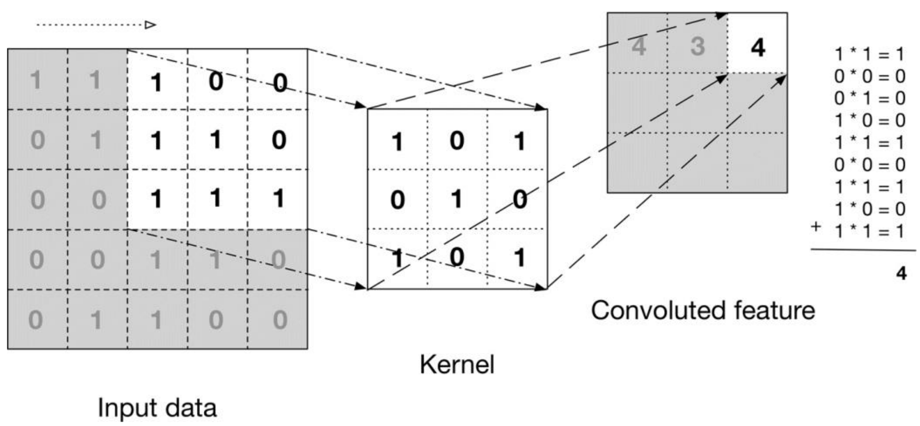


Figure.21 Showing the Convolution Operation. (Patterson & Gibson, 2017).

POOLING LAYER

In order to lessen the spatial dimension with invariable in-depth, a downsampling layer is tried after setoff layers. To fabricate an output based on the type of pooling, we customarily pertain a 2x2 size filter to the inputs. Pooling does not have any parameters—two types of pooling and average pooling. The filter bounds every sub-region, so either the maximum value or the average value is taken. Pooling layers lessens the size of the feature map depth is not decreased; only length and width are lowered. The process lessens the number of parameters and weights. It also lowers the cost of computation as well as training time. Apart from this, it also manages over-fitting.

The situation where the model reaches 100% or 99% on the training set but has 50% an average on the test data is known as over-fitting. It can be controlled by dropout layers. Dropout is the outcome of a function where an unplanned outset of activations is dropped out by setting the values to 0. It upgrades local reasoning by learning various and different depictions of patterns. The totality of values derived from the component-wise matrix multiplication of the two matrices' receptive field and the Kernel exhibits to gain elements in a feature map happens in the convolution operation. Afterward, to gain a lower dimension feature map. ReLu and max-pooling are used.

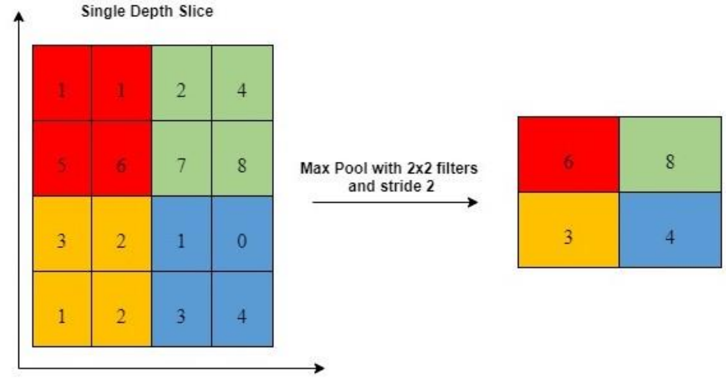


Figure.22. Pooling Operation

Figure 22 shows the example of a pooling operation max pool with 2x2 filters and stride 2 from a Single Depth Slice.

FULLY CONNECTED LAYER

This layer pinpoints the topmost characteristics that match up and agree with a device, entity, or class. For grouping images, regardless of the spatial structure of the images, input to a fully associated layer is a set of features. Two fully attached layers are used by most of the models 1 D vector is the output of the fully connected layers. It is gained by flattering the output of the final pooling layer. Process of the arranging 3 D volume into 1 D vector is called flattering.

CLASSIFIER

There are two classification types: a binary class classifier and the other is a multi-class classifier. In one binary type, the sigmoid activation function is used, whereas in the second type, a multi-class classifier function, called softmax, is used as a classifier.

MODEL SUMMARY

Table 2

NAME	TYPE	OUTPUT SIZE	PARAMETER
conv2d_1	Convolution	256 x 256 x 32	896
activation_1	ReLU	256 x 256 x 32	-
batch_normalization_1	Batch Norm	256 x 256 x 32	128
max_pooling2d_1	Pool/Max	85 x 85 x 32	-
dropout_1	Dropout	85 x 85 x 32	-
conv2d_2	Convolution	85 x 85x 64	18496
activation_2	ReLU	85 x 85x 64	-
batch_normalization_2	Batch Norm	85 x 85x 64	256

conv2d_3	Convolution	85 x 85x 64	36928
activation_3	ReLU	85 x 85x 64	-
batch_normalization_3	Batch Norm	85 x 85x 64	256
max_pooling2d_2	Pool/Max	42 x 42 x 64	-
dropout_2	Dropout	42 x 42 x 64	-
conv2d_4	Convolution	42 x 42 x 128	73856
activation_4	ReLU	42 x 42 x 128	-
batch_normalization_4	Batch Norm	42 x 42 x 128	512
conv2d_5	Convolution	42 x 42 x 128	147584
activation_5	ReLU	42 x 42 x 128	-
batch_normalization_5	Batch Norm	42 x 42 x 128	512
max_pooling2d_3	Pool/Max	21 x 21 x 128	-
dropout_3	Dropout	21 x 21 x 128	-
flatten_1	Flatten	56448	-
dense_1	Dense	1024	57803776
activation_6	ReLU	1024	-
batch_normalization_6	Batch Norm	1024	4096
dropout_4	Dropout	1024	-
dense_2	Dense	15	15375
activation_7	Softmax	15	-

Table 3: Analysis of different parameters

TOTAL PARAMETERS	TRAINABLE PARAMETERS	NON-TRAINABLE PARAMETERS
58,102,671	58,099,791	2,880

3. Results

Table 4 below shows the Several Data Split (Train, Test) and the obtained Accuracy, Precision, Recall, F1-Score, and Support at different Splits.

Table 4: Experimental results for different data split

DATA SPLIT (TRAIN, TEST)	ACCURACY (%)	PRECISION (Mean)	RECALL (Mean)	F1-SCORE (Mean)	SUPPORT (Mean)
TRAIN 10 %, TEST 90%	92.9644	0.61	0.45	0.41	2657
TRAIN 20 %, TEST 80%	93.9175	0.66	0.54	0.53	2362

TRAIN 30 %, TEST 70 %	97.5358	0.82	0.81	0.81	2067
TRAIN 40 %, TEST 60 %	96.6967	0.80	0.75	0.74	1772
TRAIN 50 %, TEST 50 %	92.7958	0.57	0.43	0.41	1476
TRAIN 60 %, TEST 40 %	96.1953	0.79	0.70	0.69	1181
TRAIN 70 %, TEST 30 %	97.1708	0.82	0.78	0.77	886
TRAIN 80 %, TEST 20 %	**98.1838**	**0.89**	**0.86**	**0.86**	**591**
TRAIN 90 %, TEST 10 %	96.5315	0.79	0.73	0.73	296

The term Accuracy is used to measure the extent of correctness in getting the expected results. We obtain the best result from the model using in Data Split (Train-80%, Test-20%), where the Accuracy is 98.18%. It also shows the mean Precision $_{(mean)}$ of all 15 classes, which is 0.89. The ratio between the True Positives with all Positives is Precision. This set of splits provides the most optimum results across all types of analysis.

As in Recall, the above table shows 0.86 Recall $_{(mean)}$ at (80, 20) Data Split. The recall is the ratio of True Positive with the True Positive + False Negative. Our model has Recall 0.86, which is 86% correctly identifying True Positives. It also measures the identification of 15 class analyses, relevant data accurately. Recall $_{(mean)\ was}$ obtained from 15 classes.

Now the F1 score $_{(mean)}$ 0.86 in (80,20) Train, Test Data Split. The F1 score maintains the balance between Precision and Recall. It is the Harmonic Mean between Precision and Recall also. Our model is showing an 86% score, which seems a good value.

The number of actual occurrences of the class in a specified dataset is called Support. The Support $_{(mean)}$ we obtained in Data Split (Train- 80%, Test-20%) is 591.

Table 5: Different Split data validation accuracy and loss

DATA SPLIT (TRAIN, TEST)	TRAINING AND VALIDATION ACCURACY	TRAINING AND VALIDATION LOSS

(10, 90)

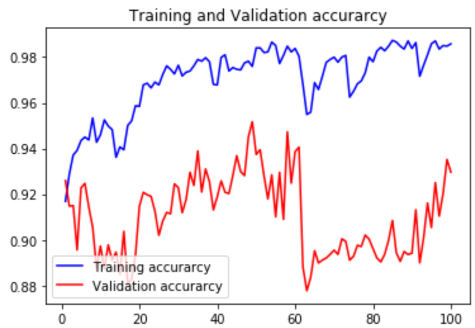


FIGURE 24

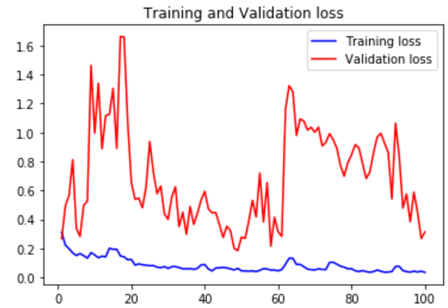
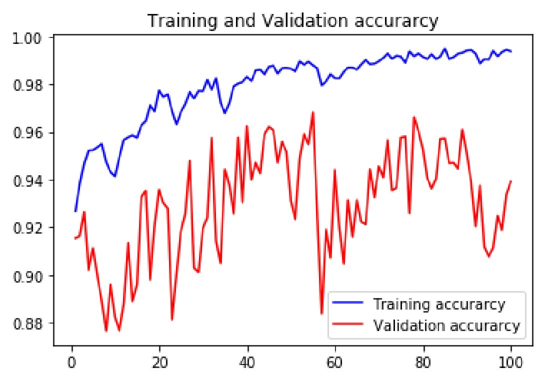


FIGURE 23

(20, 80)



FIGURE

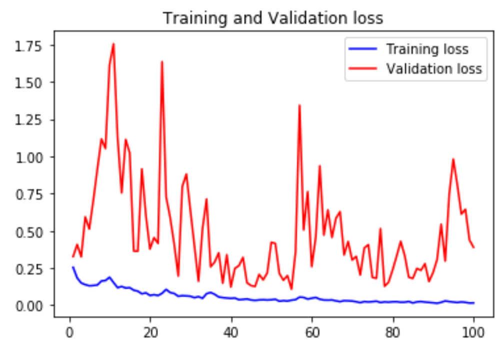
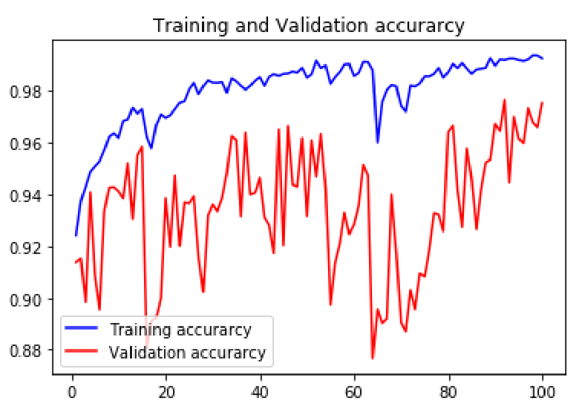


FIGURE 25

(30, 70)



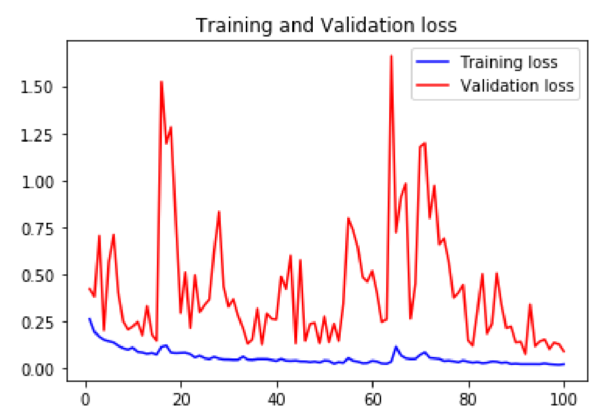
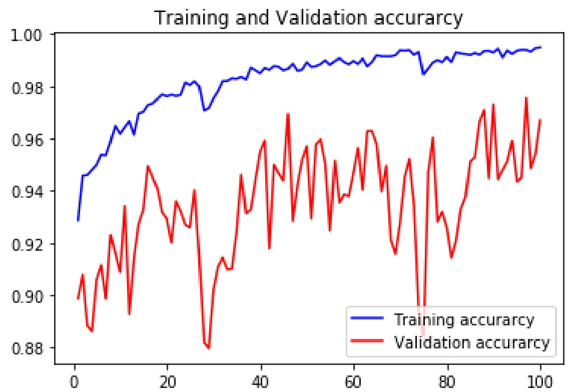


FIGURE 27

(40, 60)



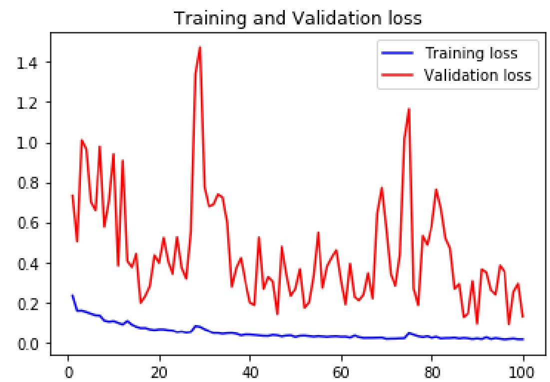
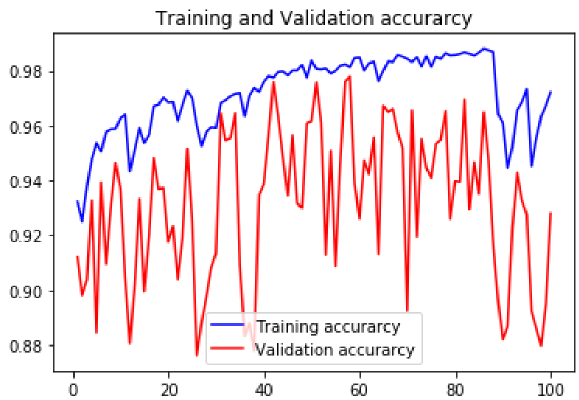


FIGURE 30

FIGURE 29

(50, 50)



FIGURE

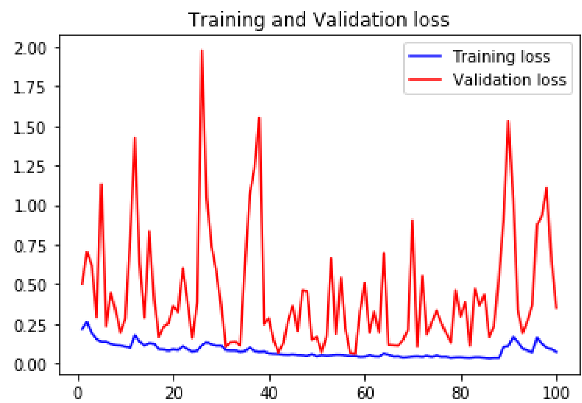
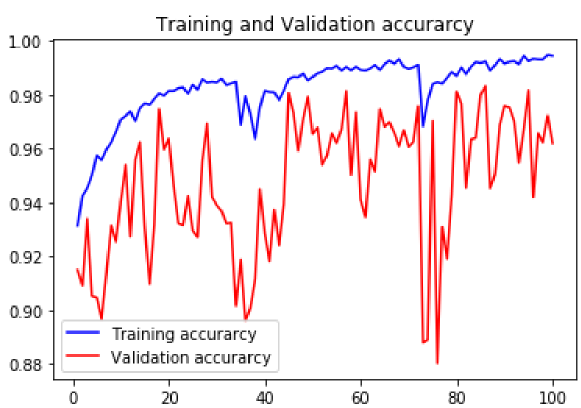


FIGURE 31

(60, 40)



FIGURE

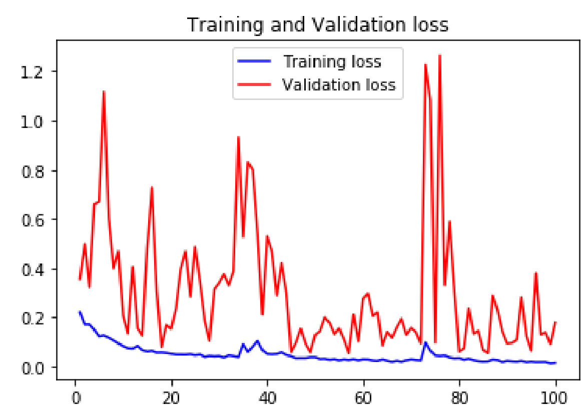


FIGURE 33

(70, 30)

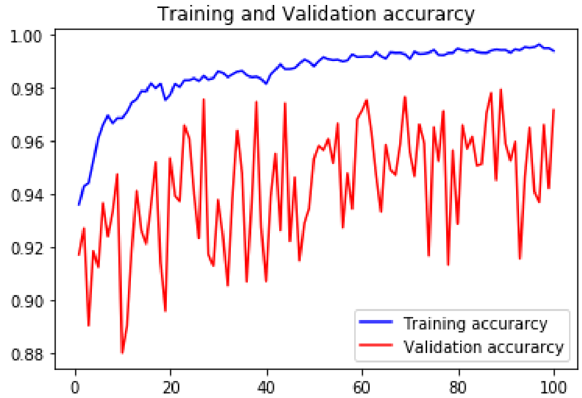
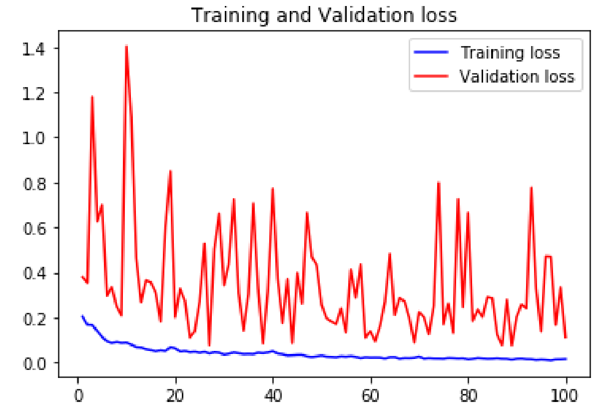


FIGURE 35



36

(80, 20)

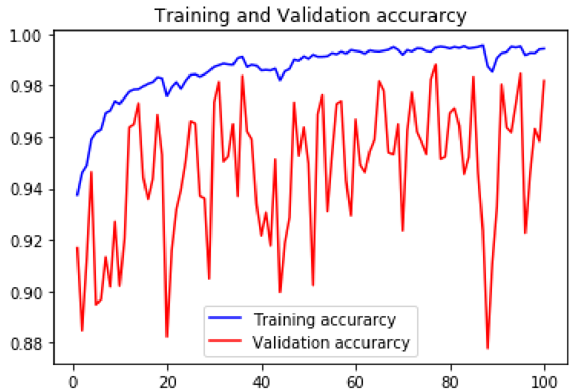
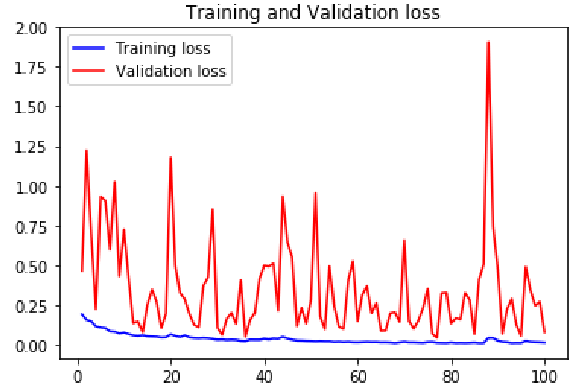


FIGURE 37



FIG

(90, 10)

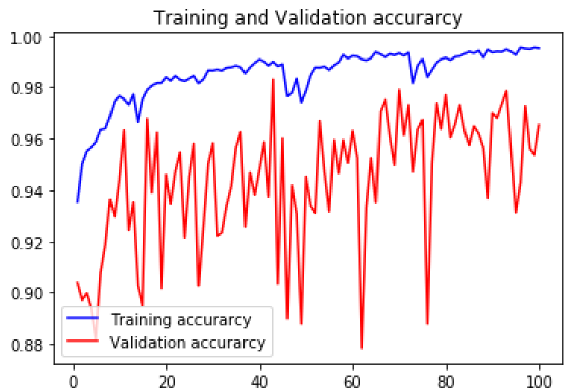
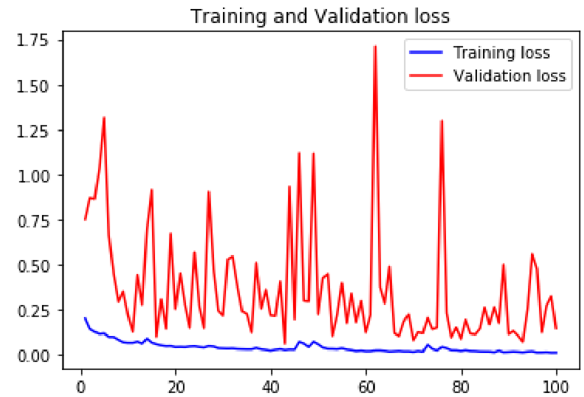


FIGURE 39



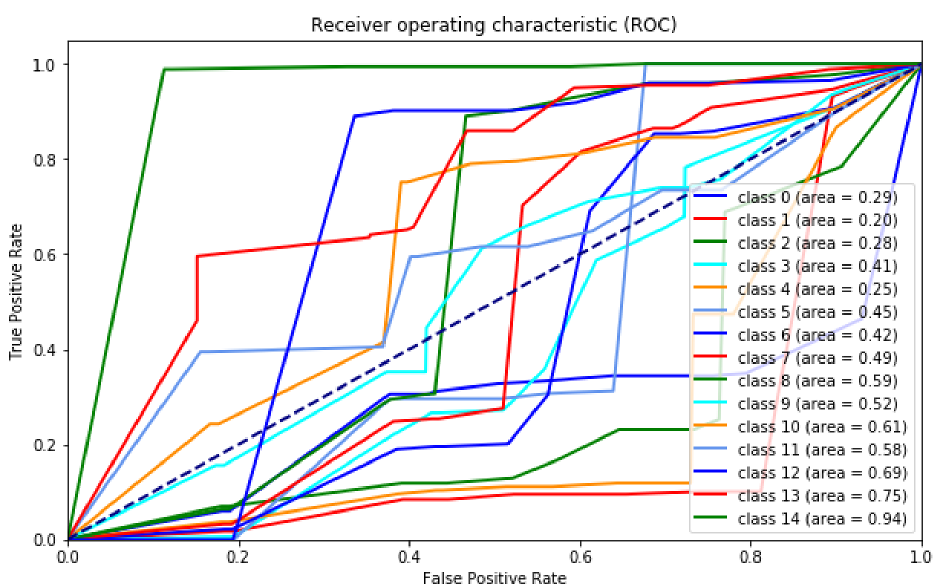
40

F

Diagrams above in the table showing Training and Validation Accuracy, also Training and Validation Loss. Now in the Training and Validation accuracy graphs, the training accuracy increases with the increase in the number of epochs. Hence, in the validation, Accuracy is shown in the graph, increasing and decreasing in the graph with each epoch. As per the graph, the validation accuracy is not continuously increasing. Such can be called the validation accuracy is quite unstable in the above figures in different Train, Test Data Splits. We obtain the best Accuracy in the (Train-80%, Test- 20%) Data Split from the above Table.

In terms of Training and Validation loss from the above graphs, the amount of validation loss is presented in each graph with the Training loss. The validation and Training loss is decreased in each increasing number of epochs. At reaching the 100^{th} epoch, the training loss is near to a minimum. As in the validation, the loss is it is unstable during the process of each epoch. In every epoch, validation loss is reaching and declining in regular frequency. So, the Training Loss is more stable as compared to the Validation loss. We obtain the least Training loss in the (Train-80%, Test-20%) Data Split from the above Table.

Table 6

DATA SPLIT (TRAIN, TEST)	RECEIVER OPERATING CHARACTERISTIC (ROC) CURVE **OBTAINED FROM SEVERAL (TRAIN, TEST) DATA SPLITS**
(10, 90)	 FIGURE 41

(20, 80)

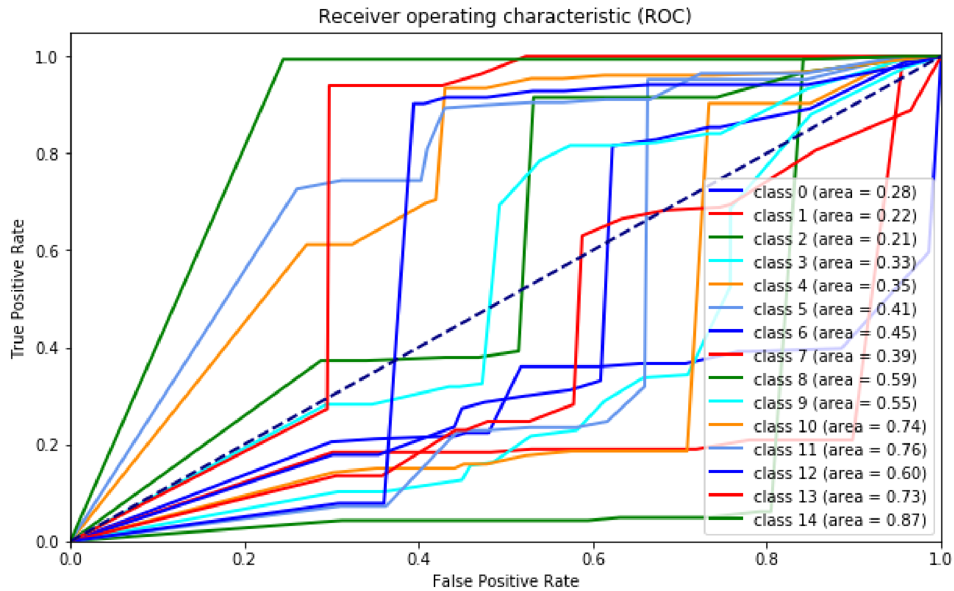


FIGURE 42

(30, 70)

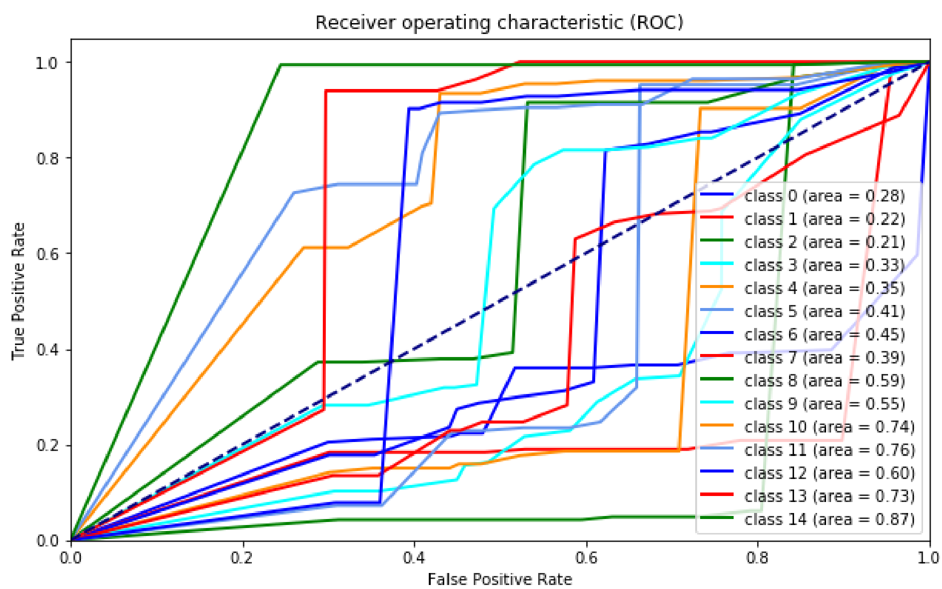


FIGURE 43

(40, 60)

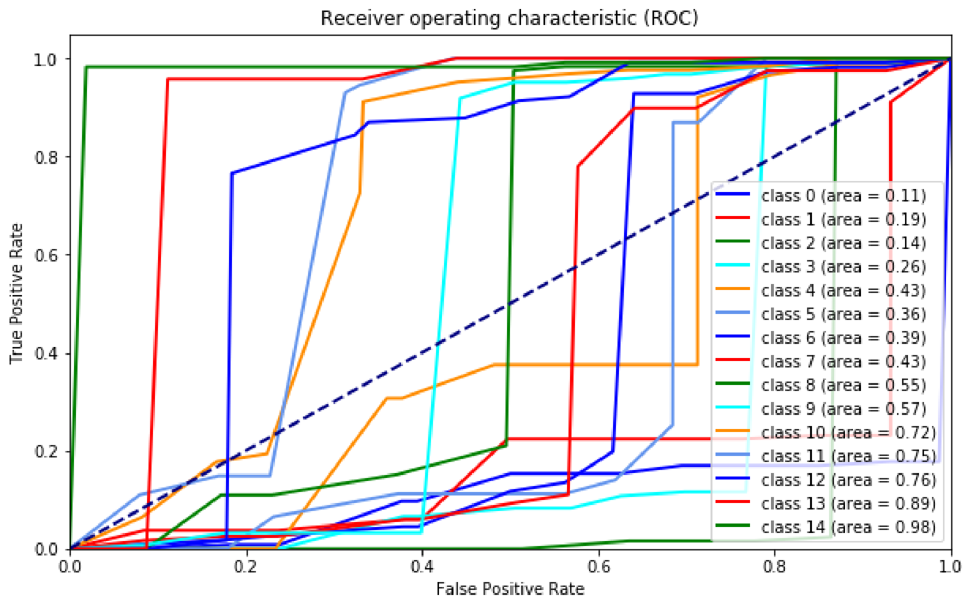


FIGURE 44

(50, 50)

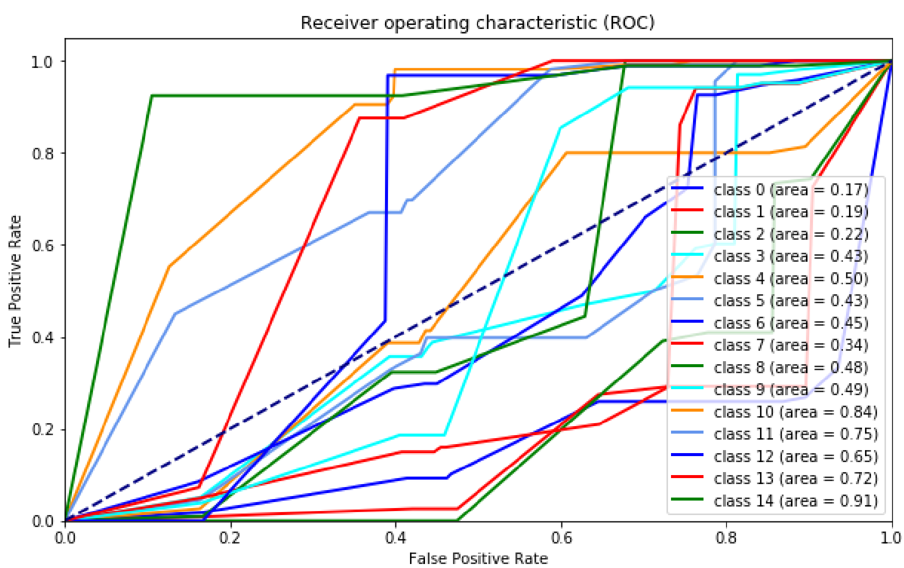


FIGURE 45

(60, 40)

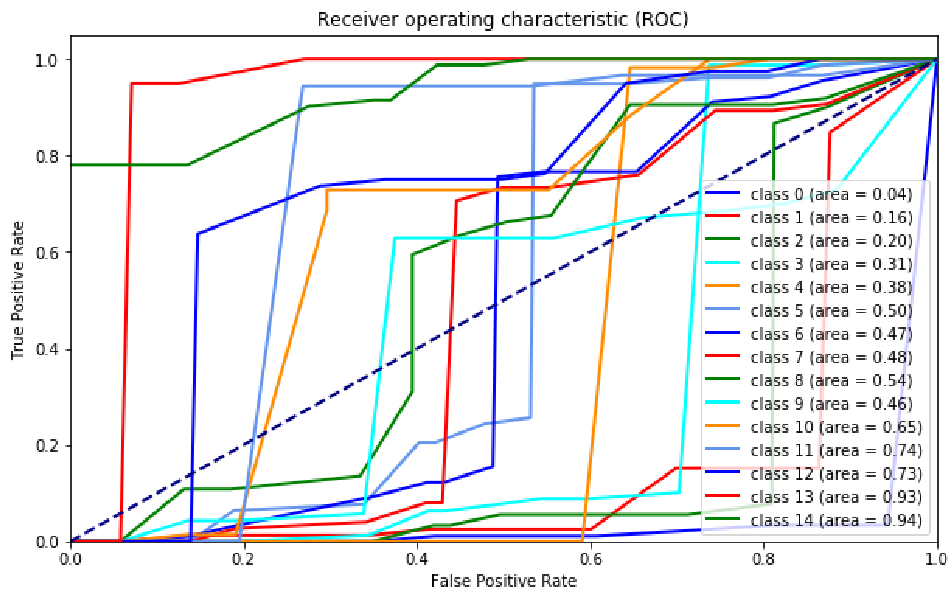


FIGURE 46

(70, 30)

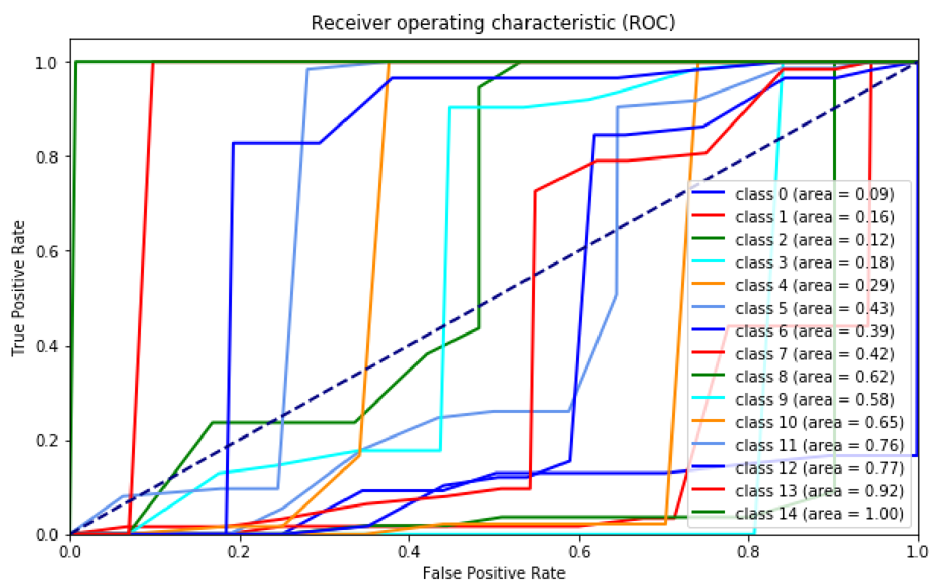


FIGURE 47

(80, 20)

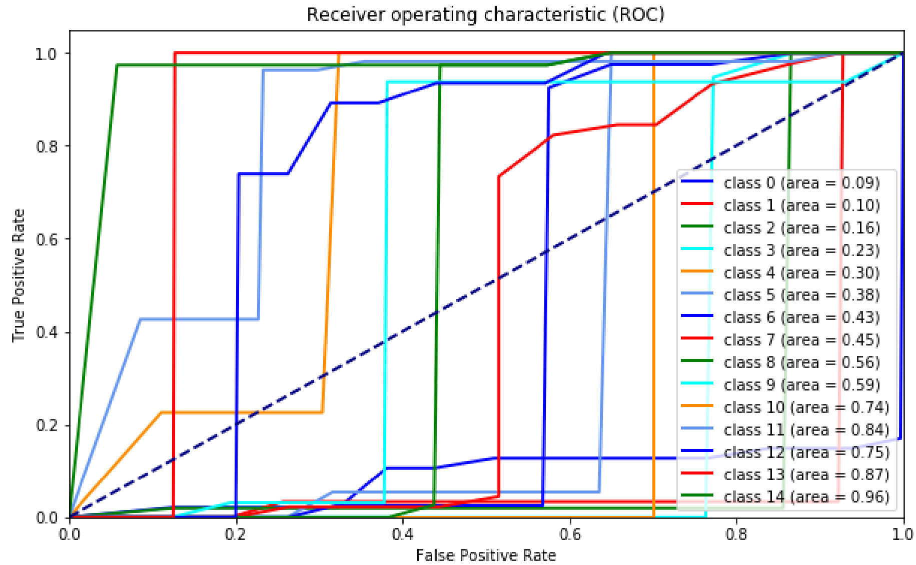


FIGURE 48

(90, 10)

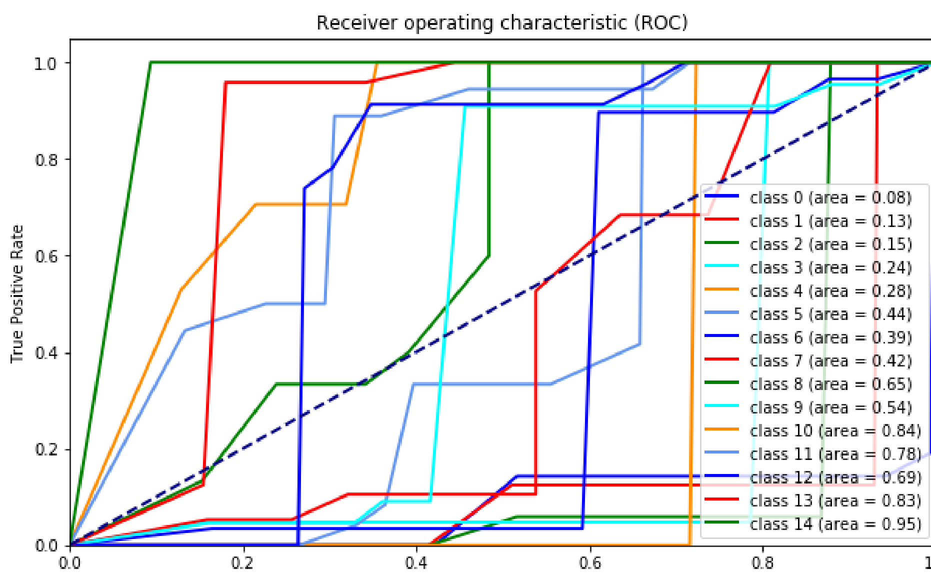


FIGURE 49

Table 6 from Figure 41 to Figure 49 showing several ROC curves obtained with different types of data splits. Among all the above graphs, if each class line in the graph is straight, it delivers a fine result. Otherwise, the Data Split combination is not perfect. The ROC curve of different classes is compared within the different data splits in which the Area Under Curve we get maximum of each class compared in (80,20) splits in Figure 48.

Table 7

CONFUSION MATRIX

OBTAINED FROM SEVERAL (TRAIN, TEST) DATA SPLITS

(Train Data -10 %, Test Data – 90%)

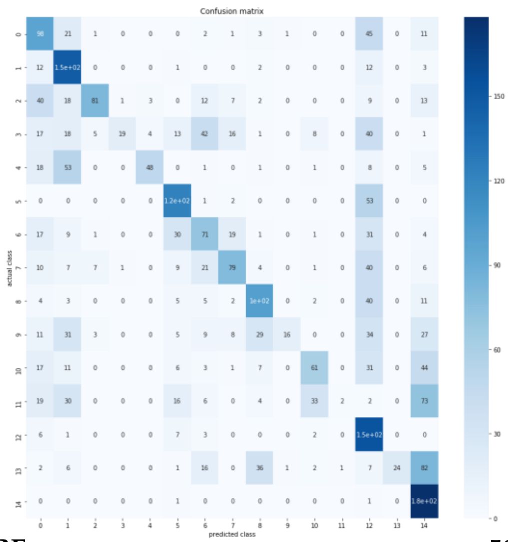


FIGURE 50

(Train Data -20 %, Test Data – 80%)

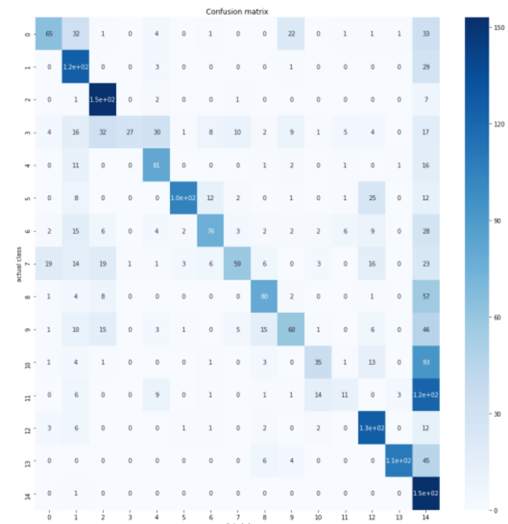


FIGURE 51

(Train Data -30 %, Test Data – 70%)

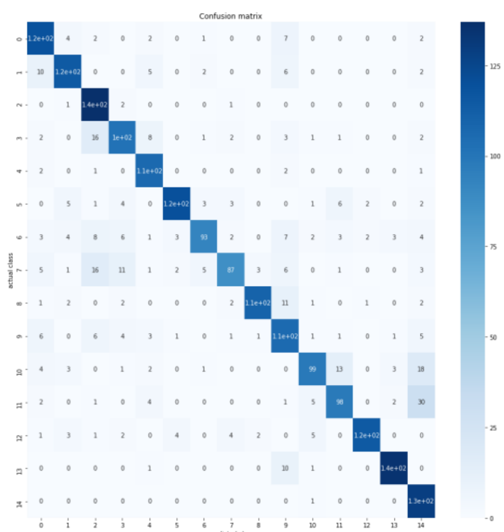


FIGURE 52

(Train Data -40 %, Test Data – 70%)

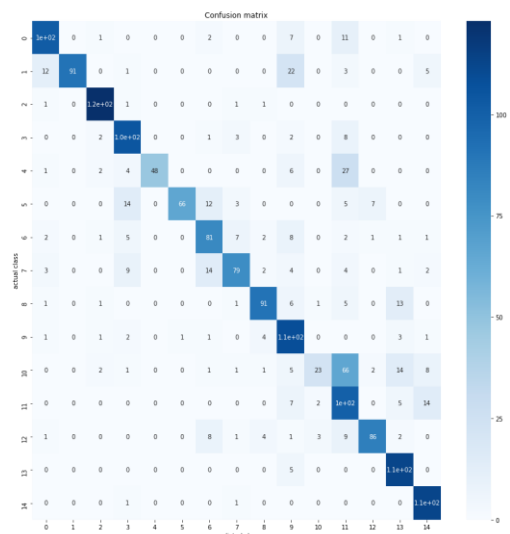


FIGURE 53

(Train Data -50%, Test Data – 50%)

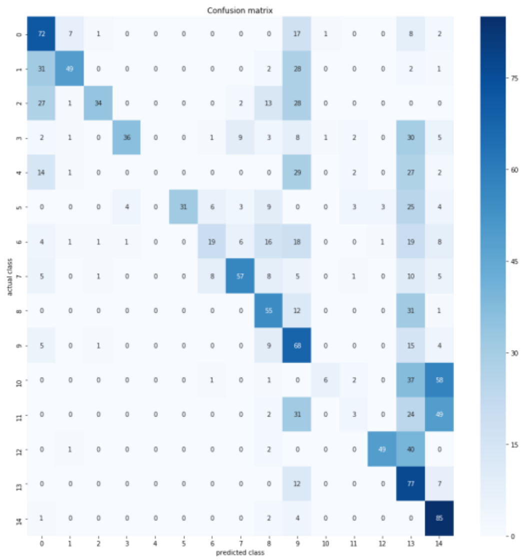
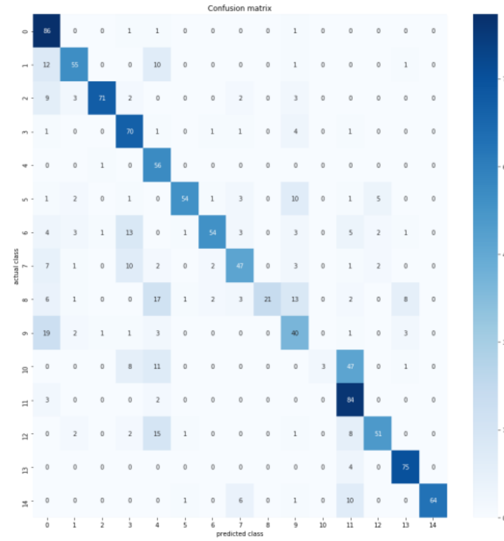


FIGURE 54

(Train Data -60 %, Test Data – 40%)

FIGURE 55



(Train Data -70 %, Test Data – 30%)

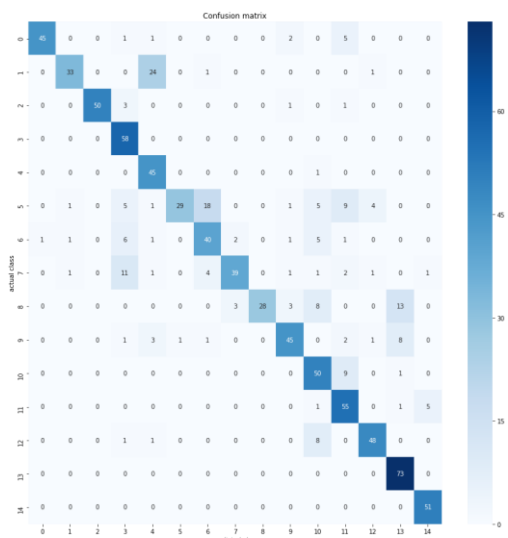


FIGURE 56

(Train Data -80 %, Test Data – 20%)

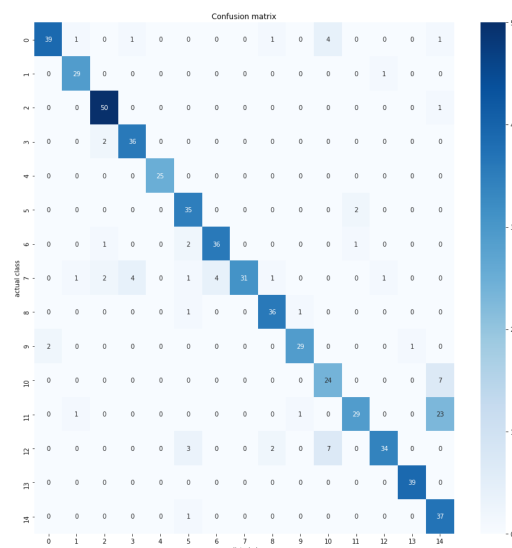


FIGURE 57

(Train Data -90 %, Test Data – 10%)

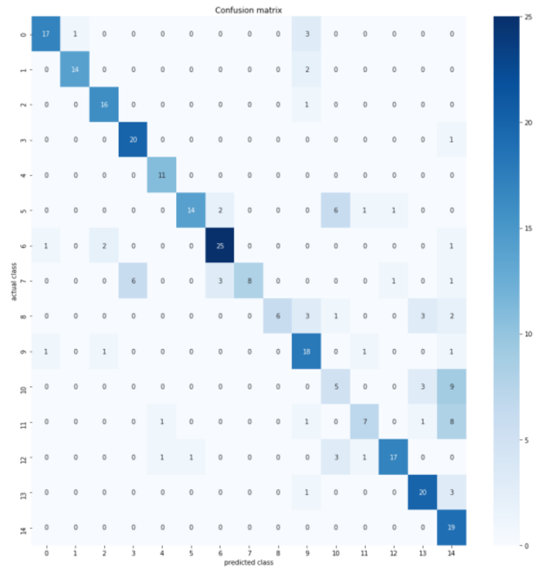


FIGURE 58

As above Table, 7 Figures from 50 to 58 showing different Confusion Matrices and the best result obtained in Data Split (Train-80%, Test-20%) in Figure 57. All other matrices show diagonal, but the values in each diagonal box are not relatable to the output as some show exponential values. Boxes in the diagonal may be darkened, but they are not showing a valid value. That is why they are not considered optimal results.

DETAILS OF THE BEST MODEL.

Table 8. TRAINING PARAMETERS OF THE MODEL.

PARAMETER	VALUE
Epochs	100
Batch Size	32
Learning Rate	1e-3
Activation in Middle Layers	Relu
Activation in Final Layer	Softmax

Table 9. PERFORMANCE METRICS OF THE BEST MODEL's CLASSES

CLASS NO.	CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT

0	**Pepper Bell Bacterial Spot**	0.95	0.83	0.89	47
1	**Pepper Bell Healthy**	0.91	0.97	0.94	30
2	**Potato Early Blight**	0.91	0.98	0.94	51
3	**Potato Late Blight**	0.88	0.95	0.91	38
4	**Potato Healthy**	1.00	1.00	1.00	25
5	**Tomato Bacterial Spot**	0.81	0.95	0.88	37
6	**Tomato Early blight**	0.90	0.90	0.90	40
7	**Tomato Late Blight**	1.00	0.69	0.82	45
8	**Tomato Leaf Mold**	0.90	0.95	0.92	38
9	**Tomato Septoria Leaf Spot**	0.94	0.91	0.92	32
10	**Tomato Spider Mites Two Spotted Spider Mite**	0.69	0.77	0.73	31
11	**Tomato Target Spot**	0.91	0.54	0.67	54
12	**Tomato Yellow Leaf Curl Virus**	0.94	0.74	0.83	46
13	**Tomato Mosiac Virus**	0.97	1.00	0.99	39
14	**Tomato Healthy**	0.54	0.97	0.69	38
	TOTAL	**0.89**	**0.89**	**0.86**	**591**

TRAINING AND VALIDATION ACCURACY

Training and validation accuracy is shown in Figure 59. When we see precisely the training accuracy compared to each epoch, the training accuracy is increasing. The test accuracy increased in the graph above 98%, which demonstrates a high accuracy. In contrast, the validation accuracy shows instability and in most of the epochs, it decreases at a very high rate.

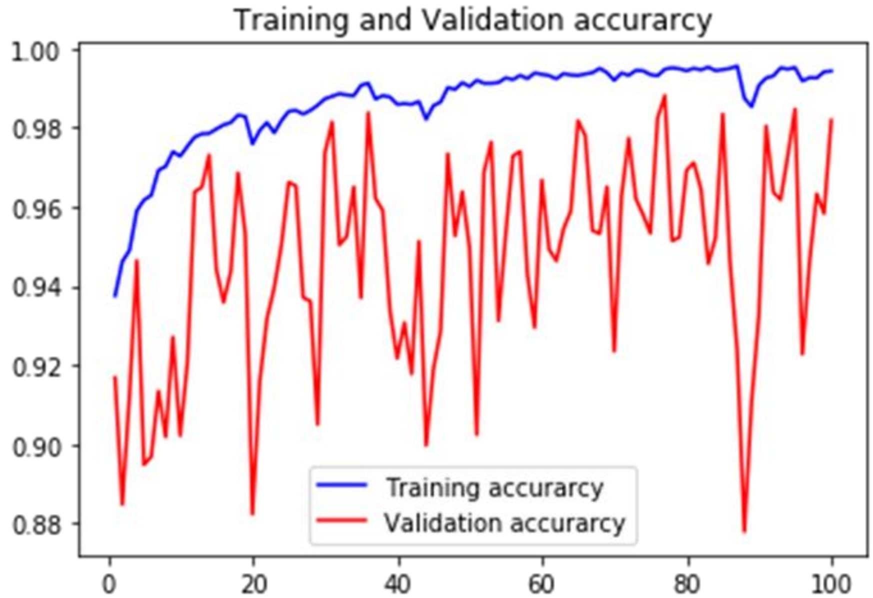


FIGURE 59 Training and validation accuracy

TRAINING AND VALIDATION LOSS

Training and Validation Loss are shown in Figure 60. In analyzing precisely, we get to know that at each epoch, the Training loss reduces. While the Validation Loss looks more unstable at some particular epoch, it rises to a high level and at the next epoch, it declined and is reduced to the previous level. The Training Loss reaches or remains nearly around 1-2 in the graph, which seems to be a good position.

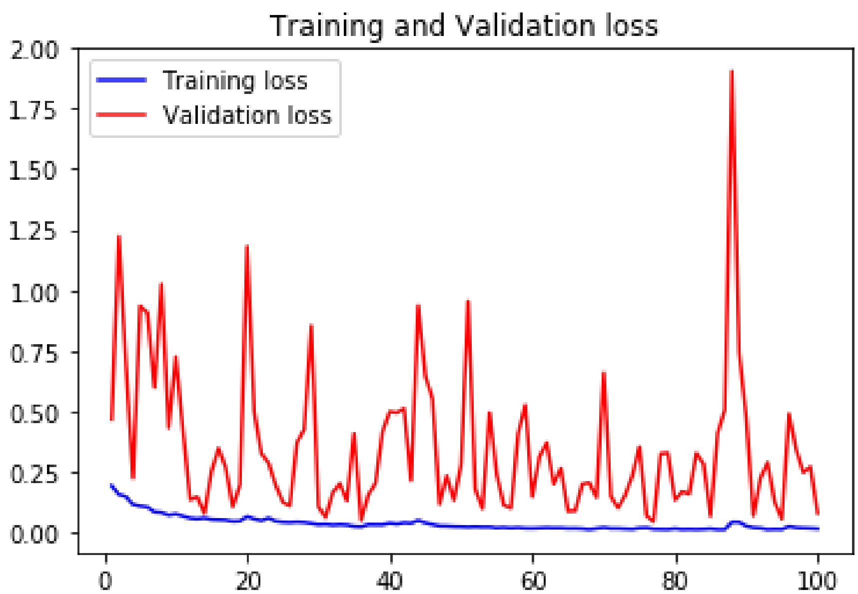


FIGURE 60: Training and Validation loss

ROC CURVE

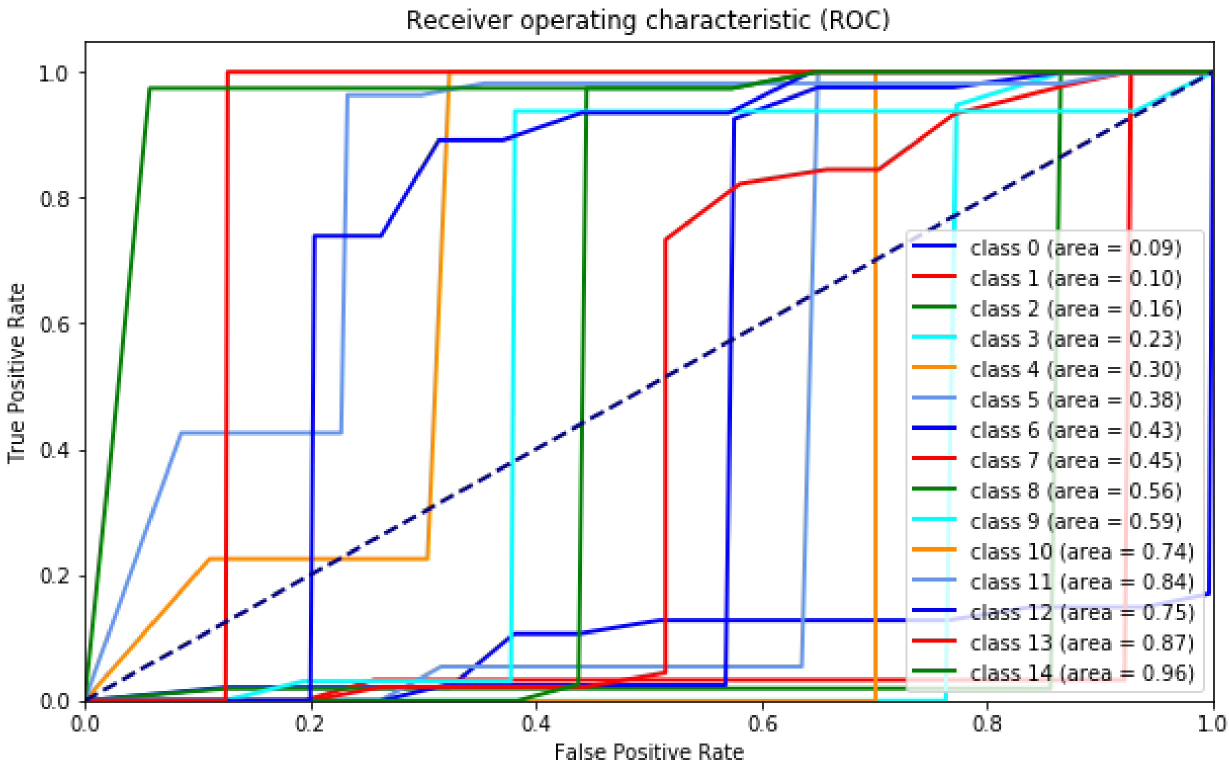
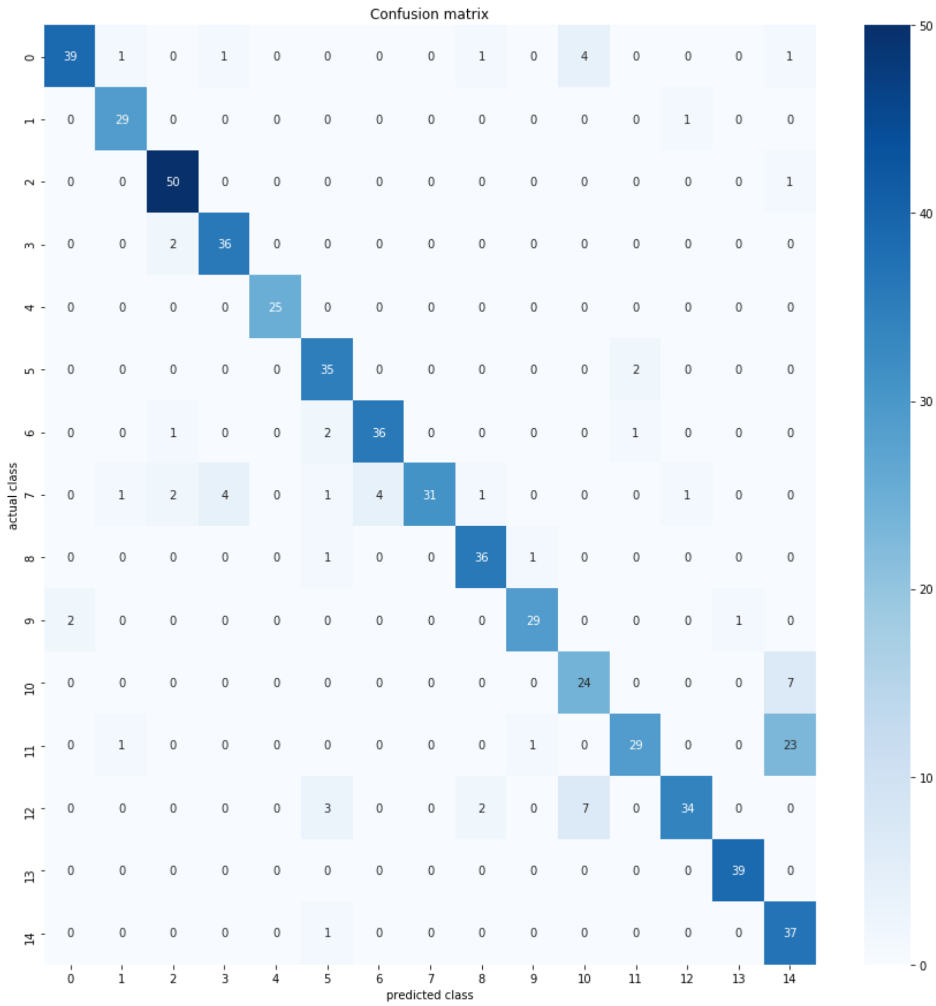


FIGURE 61. ROC Curve of the best data split.

Figure 61 shows The ROC curve obtained from the best Data Split (Train-80%, Test-20%). It shows which class is performing among all classes. Classes from 0 to 7 with less Area Under Curve (AUC) and Classes from 8 to 14 have a good Area Under Curve (AUC). As in the graph classes, 12 and 13 nearly touch the 1 point, indicating a good result.

CONFUSION MATRIX



Confusion matrix
actual class
predicted class
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
39 1 0 1 0 0 0 0 1 0 4 0 0 0 1
0 29 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 50 0 0 0 0 0 0 0 0 0 0 0 1
0 0 2 36 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 25 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 35 0 0 0 0 0 2 0 0 0
0 0 1 0 0 2 36 0 0 0 0 1 0 0 0
0 1 2 4 0 1 4 31 1 0 0 0 1 0 0
0 0 0 0 0 1 0 0 36 1 0 0 0 0 0
2 0 0 0 0 0 0 0 0 29 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 24 0 0 0 7
0 1 0 0 0 0 0 0 0 1 0 29 0 0 23
0 0 0 0 0 3 0 0 2 0 7 0 34 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 39 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 37
0 10 20 30 40 50

Figure 62 Confusion Matrix

Classifiers can be confused when faced with multiple classes of similar shapes. Infected Leaf images at different stages or against different backgrounds may also lead to the high complexity of patterns displayed in the same class, leading to lower performance. In order to visualize the estimated classification accuracy of a model, a Confusion Matrix has been used. The Figure shows the confusion matrix of the final test results. In the visualization results, the deeper the color, the higher the prediction accuracy of the model in the corresponding class. The correct predictions are on the diagonal and all incorrect predictions are off the diagonal. The study also enables us to know how to avoid confusion among classes better to improve the performance of the model. According to the 15 class analyses, Pepper Bell bacterial Spot and Potato Early Blight differ more subsequently from those other diseases. According to this Confusion Matrix, compare with other classes, the detection is more prone to distinguishing class 12 and class 13, which are Tomato Yellow Leaf Curl Virus and Tomato Mosiac Virus, respectively. Nevertheless, other classes are well differentiated. The confusion matrix explains the low recognition accuracies of several classes in our experiment. The above Confusion Matrix is obtained Data Split (Train-80%, Test-20%).

4. Related Work

(Mohanty et al. 2016) instruct a deep convolutional neural network to detect 26 diseases among 14 crop species. The perfection level is 99.35%. The model mentioned above displays the likelihood of the approach. The said model reaches the Accuracy of 31.4 % various training details necessary to enhance the common exactness. In brief, we can set forth that training deep learning models emerge clearly towards the smartphone. (Sumbwanyambe & Sibiya, 2019) gather the images of diseased leaves of maize, and a smartphone camera was applied by the CNN network. The outcome was mainly three disease detection, 1) The Northern leaf Blight, 2) Common Rust, 3) Grey leaf spot. To make it more simple, Deep CNN Neuroph Studio Framework was used as IDE. It came up with 92.85 accuracies. Dhakal & Shakya demonstrated a new model that Mr. Ashwin Dhakal introduced to detect diseases among plant leaves. Four category labels were used – Bacterial Spot, Yellow Leaf Curl Virus, Late Blight, Healthy Leaf. Several Artificial Neural Network architectures were applied and accuracy results shot up to 98.59 %. (Karol et al.,2019). show that a defense mechanism may also be used to recognize diseased plants and cure them. The database is detailedly divided. It was taken from the internet. It comprises a variety of plant diseases and is acknowledged to make a correct database. For prediction, CNN was applied. An original drone model joined with a high-resolution camera was joined to know the plant is healthy or not. The perfection rate is 78%. The system is based on python speed accuracy can be enhanced with the help of Google's GPU. Boulent et al. present an overview of 19 studies was gained. It was based on CNN and self-implemented on the identification of diseased crops. In order to judge the expectations of CNN in ecology, environment, some instructions are provided to follow.

On the other hand, (Goncharov et al. 2019) present the assembled and constructed group that may influence the authentication of the Accuracy of the neural model while we deal with the images from real life. During research work, the difficulty with the plant village images group was observed. An exceptional database of grape leaves full of four sets of images was collected. The database Deep Siamese convolutional neural network was processed to settle the difficulty of small images. The recognition of ESCA Black rot and Chloorosis disease on the grape leaves was with 90% validity. A lot of comparative outcomes and results of different models and plants were introduced. In this recognition research field, Siamese neural network in viewpoint. We are prepared to use PDDP as basic deep learning architecture. The spirit and vitality lie in probing for dissimilarities between classes, so we will compute and add more data. The quality of the databases is completely keynoted for the result, so to expand the technology system of fulfilling DB with images is correctly recognized. In the paper (Wallelign et al.,2018), CNN resuscitates the main characteristics of plant disease and it also categorizes them from images from the natural environment. The model is based on LeNet architecture. 12,673 samples of leaf images were obtained in an uncontrolled environment. The accuracy rate was 99.32%. (Purushothaman et al.,2018) focus on studying images of tomato leaves derived from plant village dataset an architecture named Alex net and VGG

16 Net furnished as input. In the categorization and accuracy importance of hyper-parameter is very few. To categorize tomato crop diseases Alex Net and VGG 16 Net are used. For this, validity for VGG 16 Net was 97.29% and for Alex Net, it was 97.49 %.

(Francis and Deisy, 2019) illustrate the leaf pictures of apple and tomato plants used to identify disease among them between healthy and infected plants by the model convolutional neural network. The model is also used to categorize and tabulation also. This model is comprised of four convolutional layers; these are applied to recognize diseases. The authenticity rate is 88.7%. This model is applicable in any electronic set. The planning has been done to use more export models. Tulshan and Raul et al. (2019) show that the farmers, agriculture experts, and the government were worried about plant leaves disease. In order to solve this, a plant detection process was applied to image preprocessing. Image Segmentation and Feature Extraction are some key steps to the procedure. The Accuracy and authenticity were about 99% in forecasting diseases in plant leaves. We were also done a test on an algorithm on five affecting plant diseases. The success rate is about 99%. They are looking forward to showing more validity and authenticity in this field.

(Militante et al. 2019) assert that the current progress is the vision of the computer is to use a camera to catch the images and pictures of diseased leaves. This method was developed by Deep Learning to identify infection in leaves of the plant such as apple, corn, grapes, potato, sugarcane, and tomato images of 35,000 healthy and diseased plant leaves were collected to recognize plant disease. For this purpose, a deep learning model was developed by Research fellows. The Accuracy of this model was 96.5 % and 75 epochs were used in this method. One method shows 100% accuracy of healthy and diseased plants. The second method shows 99.62% accuracy of leaves infected with the disease, namely late blight and early blight diseases. The third method displays an image of a grape plant showing images of infected leaves with late blight disease and black rot disease. Farmers can use this model to identify plant diseases. Its accuracy rate is 96.5%. (Kosamkar et al. 2018) show the current paper suggests working preprocessing feature extraction by using Tensor Flow technology, in which this system suggests pesticide. The topmost validation accuracy is about 90% for those using Tensor Flow. A convolutional neural network with a five-layer model with a four and three-layer model is used to categorize diseases in leaf images. With the five-layer model, the topmost validity and Accuracy were 95%. In contrast, Liu et al. (2019) confirm that several diseases spoil the apple crop, namely grey spot, mosaic, Brown spot, and Alter Naria leaf spot. The current paper is based on the model Google Net Inception structure and Rainbow Concatenation. A new apple leaf disease identification model using DEEP-CNN has been introduced. The model INAR-SSD is used to identify five common apple leaf diseases. The detection result was 78.80%. The INAR-SSD model can identify the disease of the primary stage with the correct time and topmost Accuracy based on an improved convolutional neural network for apple leaf; a real-time identification approach has been introduced. Hassan et al. (2021) came up with a transfer learning approach with a deep convolutional neural network to detect 14 different plant diseases and 38 categorical diseases. They introduce depth-separable convolution instead of a normal convolution, which reduces the cost of computation and parameter numbers. Their models achieved a rate of accuracy of 98.42%, 99.11%, 97.02%, and 99.56% using InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0, Models respectively, which are more efficient than traditional approaches, also this model is promising to implement in real-time agricultural systems. Chowdhury et al. (2021) proposed a deep learning model based called EfficientNet and tested 18,161 images of tomato leaves for disease. They distributed the model into two segments U-net and Modified U-net. EfficientNet-B4 acquired accuracy of 99.89% for ten-class classification.

5. Conclusions

To determine and segregate Tomato, Pepper Bell, and Potato Plant diseases, Convolutional Neural Network is used in this study. A trained network was used with images derived from the Plant Village Dataset of Kaggle and it reached allocation and segregation potential at 98.18 %. This is crystal clear that CNN can draw out the main features and characteristics essential for plant disease classification in the neural environment. We confirm that this experiment also shows when the dataset is very small, implementing data augmentation on the training set gears up the presentation and execution of the network. The impact of dropout to defeat over-fitting was also certified. This CNN was executed to segregate whether a leaf is diseased or healthy or not. The minimum number of parameters was 58K. Creating and training a CNN model from scratch was a monotonous procedure. In terms of our future work, the plan will make our work portable to mobile or any other implanted or inserted applications to provide a higher Accuracy with the minimum size and complexity.

Acknowledgment

This work is supported by VC Research (VCR 000148) for Prof. Chang.

References

1) Boulent J, Foucher S, Théau J, St-Charles PL. Convolutional Neural Networks for the Automatic Identification of Plant Diseases. Front Plant Sci. 2019;10:941. Published 2019 Jul 23. doi:10.3389/fpls.2019.00941.

2) Dhakal, Ashwin Prof. Dr. Subarna Shakya, (2018). Image-Based Plant Disease Detection with Deep Learning. International Journal of Computer Trends and Technology, 61(1), 26-29.

3) Francis, Mercelin, and C. Deisy. “Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks — A Visual Understanding.” 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN) (2019): 1063-1068.

4) Goncharov P., Ososkov G., Nechaevskiy A., Uzhinskiy A., Nestsiarenia I. (2019) Disease Detection on the Plant Leaves by Deep Learning. In: Kryzhanovsky B., Dunin-Barkowski W., Redko V., Tiumentsev Y. (eds) Advances in Neural Computation, Machine Learning, and Cognitive Research II. NEUROINFORMATICS 2018. Studies in Computational Intelligence, vol 799. Springer, Cham. https://doi.org/10.1007/978-3-030-01328-8_16.

5) Jiang. P, Y. Chen, B. Liu, D. He, and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 59069-59080, 2019, doi: 10.1109/ACCESS.2019.2914929.

6) Kosamkar. P. K, V. Y. Kulkarni, K. Mantri, S. Rudrawar, S. Salmpuria, and N. Gadekar, "Leaf Disease Detection and Recommendation of Pesticides Using Convolution Neural Network," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697504.

7) Militante, Sammy V., Bobby D. Gerardo, and Nanette V. Dionisio. "Plant Leaf Detection and Disease Recognition using Deep Learning." 2019 IEEE Eurasia Conference on , Communication, and Engineering (ECICE) (2019): 579-582.

8) Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.* 7:1419. doi: 10.3389/fpls.2016.01419.

9) Patterson. Josh,; Adam Gibson. (2017). Deep Learning: A Practitioner's Approach. O'Reilly Media; 1 edition.

10) Plant Disease Detection using CNN & Remedy Adnan Mushtaq Ali Karol1 , Drushti Gulhane2 , Tejal Chandiwade3 BE Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India1 B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India2 B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India3.

11) Rangarajan. Aravind Krishnaswamy, Raja Purushothaman, Aniirudh Ramesh, Tomato crop disease classification using pre-trained deep learning algorithm, Procedia Computer Science, Volume 133, 2018, Pages 1040-1047, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.07.070.

12) Sibiya, M.; Sumbwanyambe, M. A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks. AgriEngineering 2019, 1, 119-131.

13) Tulshan, Amrita S. and Nataasha Raul. "Plant Leaf Disease Detection using Machine Learning." 2019 10th International Conference on Computing, Communication, and Networking Technologies (ICCCNT) (2019): 1-6.

14) Wallelign.S, Mihai Polceanu, Cédric Buche. Soybean Plant Disease Identification Using Convolutional Neural Network. FLAIRS-31, May 2018, Melbourne, United States. pp.146--151. ⟨hal-01807760⟩.

15) Hassan, S.M.; Maji, AK; Jasi' nski, M.; Leonowicz, Z.; Jasi' nska, E. Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. Electronics 2021, 10, 1388. https://doi.org/10.3390/ electronics10121388.

16) Chowdhury, M.E.H.; Rahman, T.; Khandakar, A.; Ayari, M.A.; Khan, A.U.; Khan, M.S.; Al-Emadi, N.; Reaz, M.B.I.; Islam, M.T.; Ali, S.H.M. Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques. AgriEngineering 2021, 3, 294–312. https://doi.org/10.3390/ agriengineering3020020.