**Leaf Disease Detection using**

**Deep Learning**

Project report submitted for

**Solving for India Hackathon**

By

**Lekhana**

**Shubham Raj Sahu**

**Tavisha Thaware**



**Dr. Shyama Prasad Mukherjee**

**International Institute of Information Technology, Naya Raipur**

**(A Joint Initiative of Govt. of Chhattisgarh and NTPC)**

**Email:** [**iiitnr@iiitnr.ac.in**](mailto:iiitnr@iiitnr.ac.in)**, Tel: (0771) 2474040, Web:** [**www.iiitnr.ac.in**](http://www.iiitnr.ac.in)

**Table of Contents**

Title  Page No.

**TABLE OF CONTENTS 2**

**LIST OF FIGURES 3**

**CHAPTER 1 INTRODUCTION 4**

* 1. Motivation 4
  2. Objective 4

**CHAPTER 2 METHODOLOGY 5**

2.1 Dataset Description 5

* 1. Approach 5

2.2.1. VGG-16 6

2.2.2. Inception V3 7

* 1. Experimentation 8
  2. Measurement of Performance 9

**CHAPTER 4 RESULTS 10**

**CHAPTER 5 CONCLUSIONS 13**

**REFERENCES 14**

**List of All Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Title** | **Page Number** |
| 1 | Glimpse of dataset collected | 5 |
| 2 | Flowchart of project approach | 6 |
| 3 | Architecture of VGG-16 | 6 |
| 4 | Architecture of Inception V3 | 8 |
| 5 | Diseased Tomato leaf | 10 |
| 6 | Diseased cauliflower leaves and its features extracted after 1st and last hidden layer | 10 |
| 7 | Diseased mango leaves and its features extracted after 1st and 4th hidden layer | 10 |
| 8 | Accuracy results | 11 |
| 9 | Bar plot of accuracy | 11 |
| 10 | F1 score results | 11 |
| 11 | Bar plot of F1 score | 12 |

**CHAPTER 1. INTRODUCTION**

**1.1 Motivation**

India is a mass producer of crops which leads to the high dependency of the Indian economy on agriculture. Therefore, it is essential to consider the protection and necessary cure of different plants and detect any disease early to eliminate the possibility of crops getting ruined. To see the disorders early, using automated techniques is a novel approach. The existing methods indicate manual methods to detect leaf diseases which use a large team of agricultural experts and continuous analysis of crops. This becomes a tedious and time-consuming task as analysing large farms manually obviously takes a lot of time and cost and, most importantly, gives less accurate results.[1]

**1.2 Objective**

The objective of this project is to develop a model which can predict if the plant is diseased or healthy, given an image of the leaf of the plant and to deploy this deep learning based model using a website.

**CHAPTER 2. METHODOLOGY**

**2.1 Dataset description**

The dataset that we’ve collected contains leaf images of mainly three plants - cauliflower, tomato and mango. We are analysing a total of 37950 images of leaves which include both healthy and disease affected leaves. The dataset contains a spread of 20 class labels assigned to them, which are a crop-disease pair, and we here are trying to estimate the crop-disease pair given just the image of leaves. The images are resized to 150x150 pixels and both the model optimization and predictions on these downscaled images for all the below mentioned approaches. The dataset was augmented later to create more images of different rotation range, shift range, shear range, zoom range, brightness range etc. Figure 3 given below shows a glimpse of the dataset collected by us.

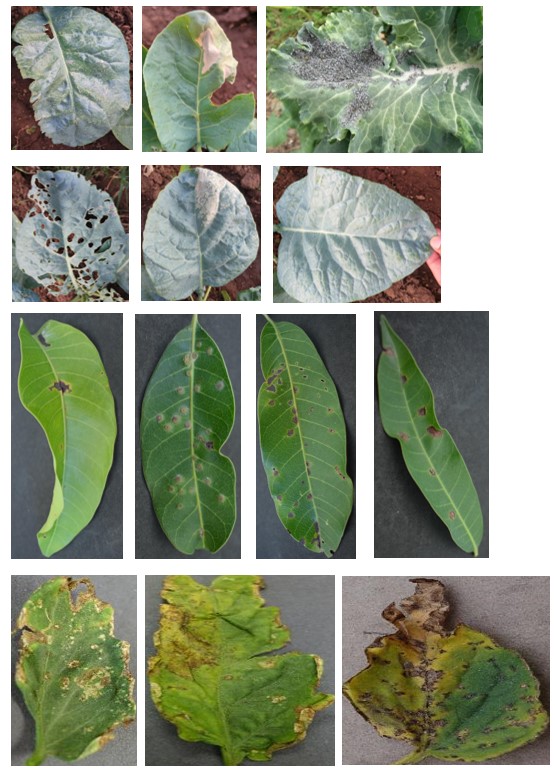


Fig 1. Glimpse of the dataset collected

**2.2 Approach**

To estimate the classification of dataset into crop-disease pairs, we are using two deep convolutional neural network architectures, VGG-16 and Inception v3 (GoogleNet), both of which were designed at the competition of “The ImageNet Large Scale Visual Recognition Challenge” ([ILSVRC](http://www.image-net.org/challenges/LSVRC/)) for the ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. Following is a flowchart of our approach towards this project.

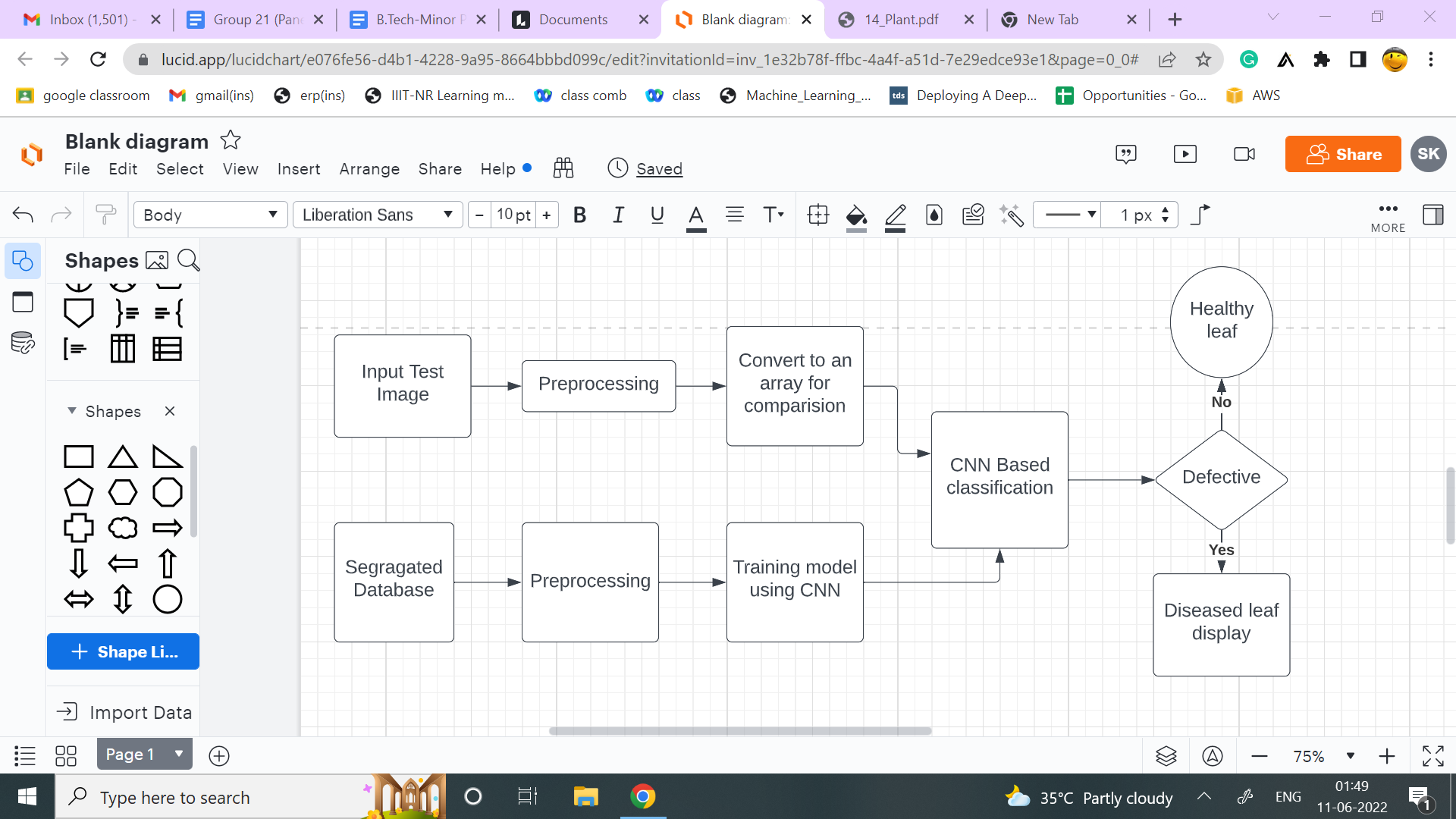


Fig 2. Flowchart of project approach

**2.2.1 VGG-16**

The VGG-16 was designed by Karen Simonyan and Andrew Zisserman of Oxford University in 2014[10]. It achieved 92.7% of top-5 test accuracy in ImageNet. The architecture of VGG-16 is depicted below in Figure 3.



Fig 3. Architecture of VGG-16[11]

The VGG-16 model consists of 13 convolutional layers, 5 max-pooling layers and 3 fully connected layers. As the number of layers having tunable parameters is 16(13 convolutional and 3 fully connected layers), hence the name VGG-16. The first convolutional takes a fixed size input of image - 224x224x3 (RGB). The first two convolutional layers of the VGG-16 model contain 64 channels of 3x3 filter size and same padding followed by a max pooling layer of stride(2,2) which is the same as the previous layer. Next there are 2 convolutional layers of filter size (3,3) and 256 filters. Following this are 2 sets of 3 convolution layers and a max pool layer, each having 512 filters of (3,3) size with the same padding. The image received after passing through all these layers is then passed to the stack of two convolution layers. In these convolution and max pooling layers, the filters are of the size 3x3. We receive a feature map following these layers, the feature map is of dimension (7,7, 512). We next flatten this feature map to make it a (1,25088) feature vector. Following this are 3 fully connected layers, of which the first layer gives output of a (1,4096) vector taking input from the previous feature vector. Similarly, the second layer also gives output of a vector of size (1,4096). The third layer output is passed to the softmax layer to normalize the classification vector. ReLU is used as the activation function in all the hidden layers. ReLU is more efficient than other functions as it results in faster learning and it also decreases the likelihood of vanishing gradient problem.[10]

**2.2.2 INCEPTION V3**

The Inception V3 is an image recognition model which is a superior version of the basic model Inception V1. The Inception V3 model was introduced as GoogleNet in 2014 and was developed by a team at Google.(Szegedy, et. al.)[12] This module has a total of 42 layers and a lower error rate than its predecessors. The improvements done on this model includes usage of label smoothing, factorized 7x7 convolutions, and an auxiliary classifier. To capture a variety of features in parallel, the inception module uses parallel 1x1,3x3 and 5x5 convolutions and a max-pooling layer. 1x1 convolutions are added before the previously mentioned 3x3, 5x5 convolutions and also after the max-pooling layer in order to reduce the dimensionality. After all this, a filter concatenation layer concatenates outputs of all these parallel layers.[13]. Following figure depicts the architecture of the Inception V3 model.

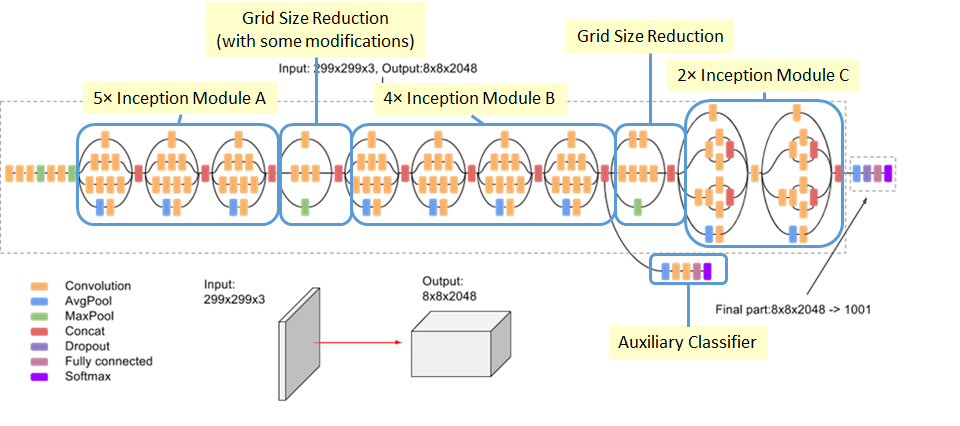


Fig 4. The architecture of Inception V3, (Batch Norm and ReLU are used after Conv)[14]

The improvements made on the Inception V3 model are as follows, Factorized convolutions - which reduces the no. of parameters involved in a network and in turn helps in reducing the computational efficiency, and also checks on the network efficiency; Smaller convolutions leads to faster training by replacing them in place of bigger convolutions; Auxiliary classifier in Inception V3 acts as a regularizer. A small CNN is inserted between layers during training as an auxiliary classifier, and the loss it suffers is added to the main network loss.[15]

**2.3 Experimentation**

Throughout all of our experiments, the dataset was used in two ways.

1. Firstly, we use the dataset in its original form, that is, in colour version.
2. After running all the experiments in this version, the dataset, next, was experimented in the grey-scale version and all the experiments were performed with this version as well.

The goal of this type of experimentation with multiple versions of the dataset is to see if the model is learning the concept of plant diseases or merely learning the dataset's inherent biases.

In total we have 8 experimental configurations, which vary on the following parameters -

Choice of CNN model

i. VGG-16

ii. Inception V3

Choice of dataset version

i. Colour

ii. Grey-scale

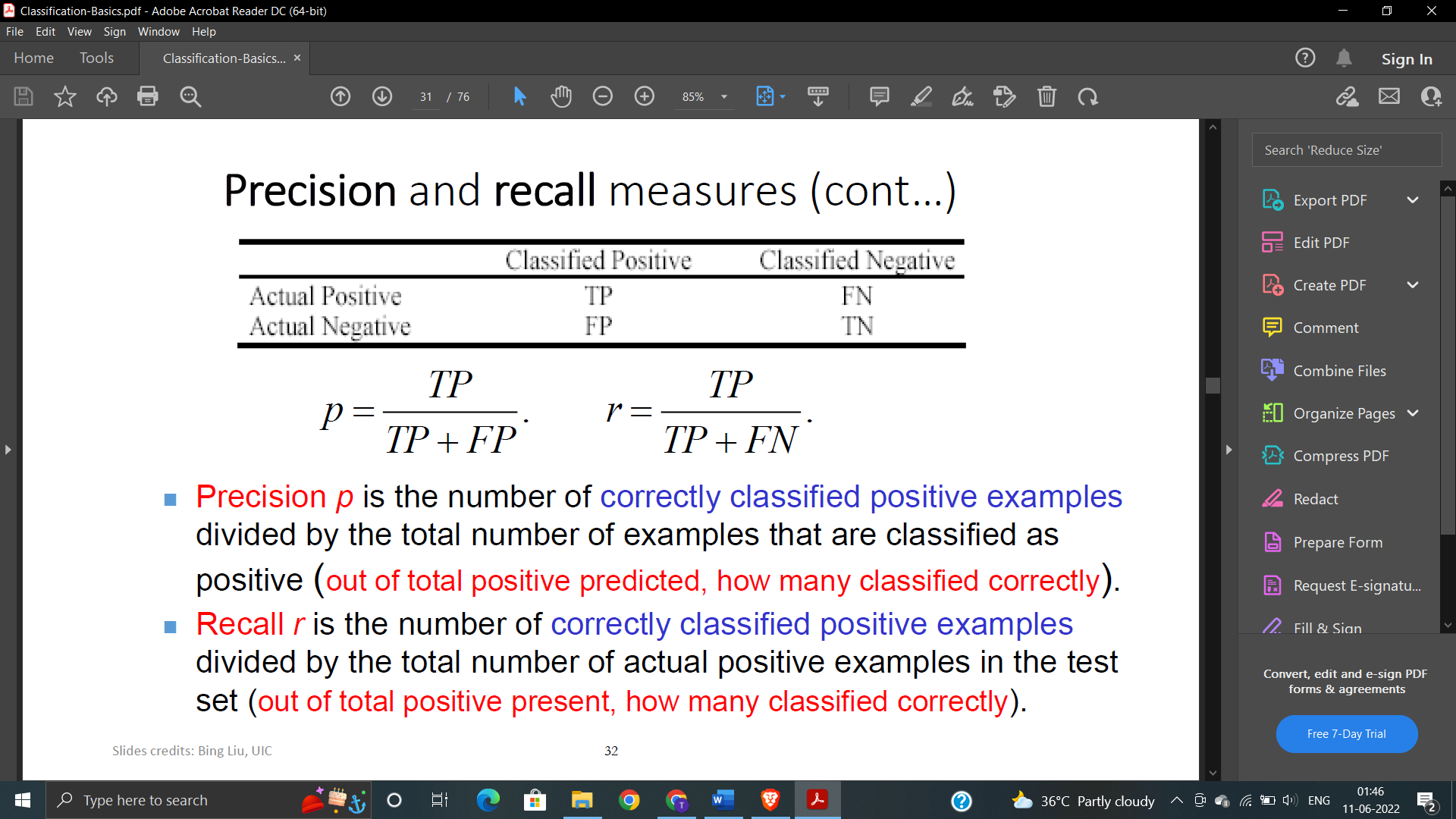
Choice of training testing set distribution

i. Training: 80% & Testing: 20%

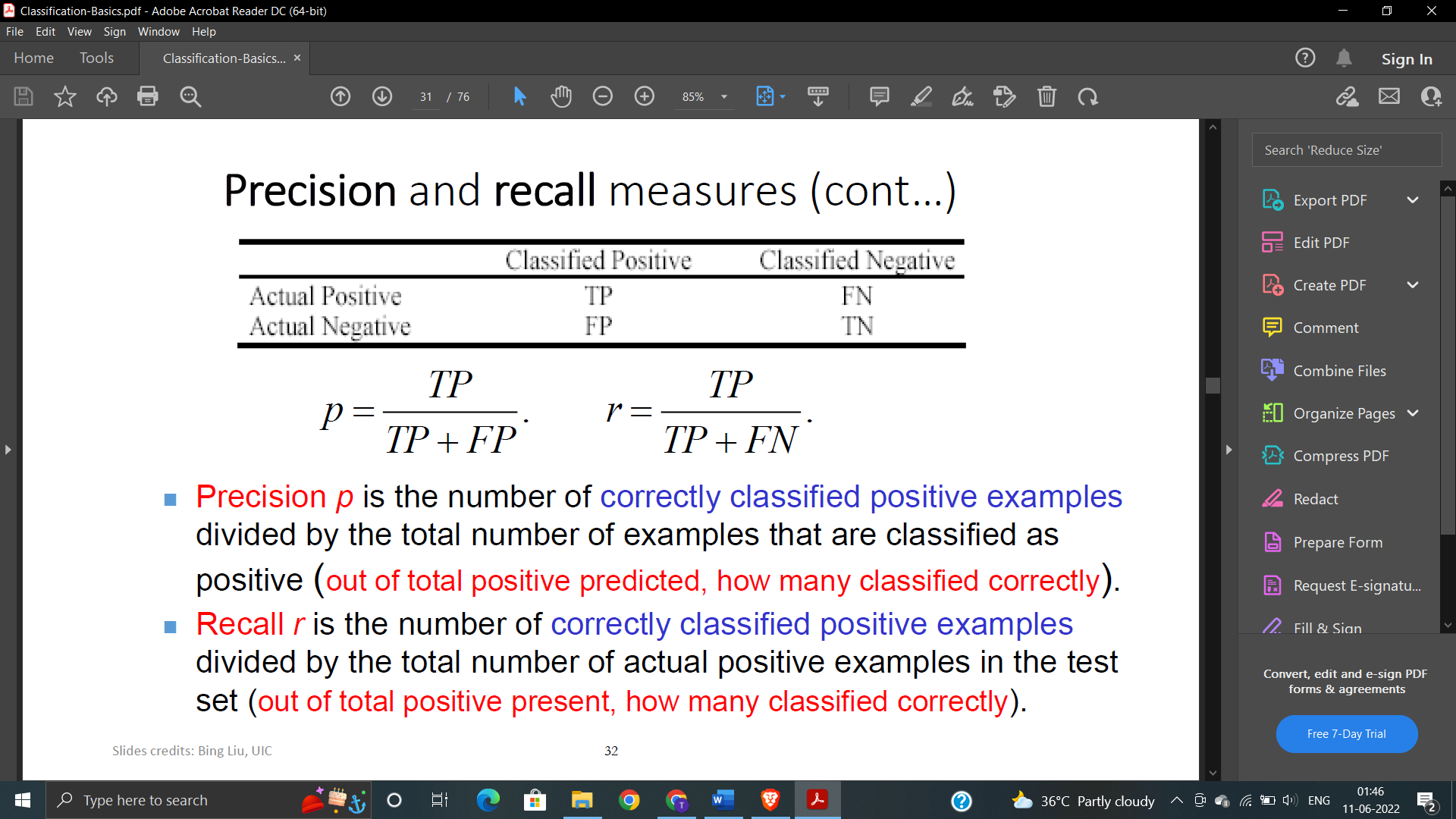
ii. Training: 60% & Testing: 40%

**2.4 Measurement of Performance**

To check the performance of the above-mentioned approaches, the dataset has been split into training and testing sets(80%-20% and 60%-40% respectively). Over these dataset splits, the deep CNN based models are run. Following that, at the end of each epoch, we compute mean precision, mean recall, mean F1 score, and overall accuracy and loss. [16]

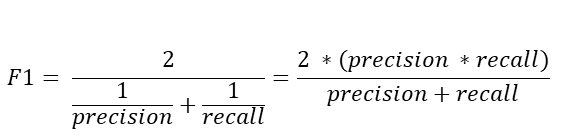
Mathematical equations representing:

1. Precision =

Precision p is the number of correctly classified positive examples divided by the total number of examples that are classified as positive (out of total positive predicted, how many are classified correctly).

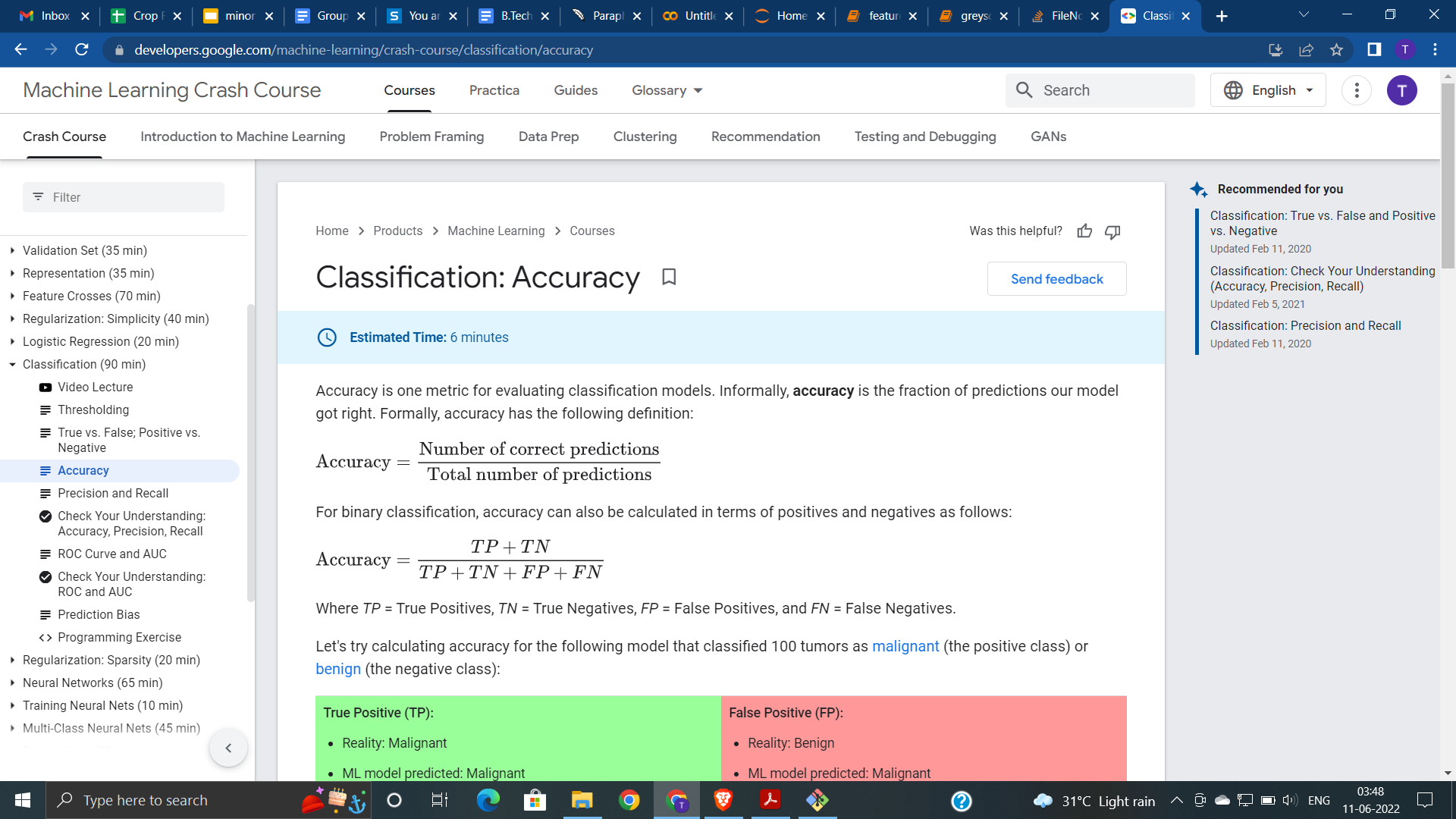
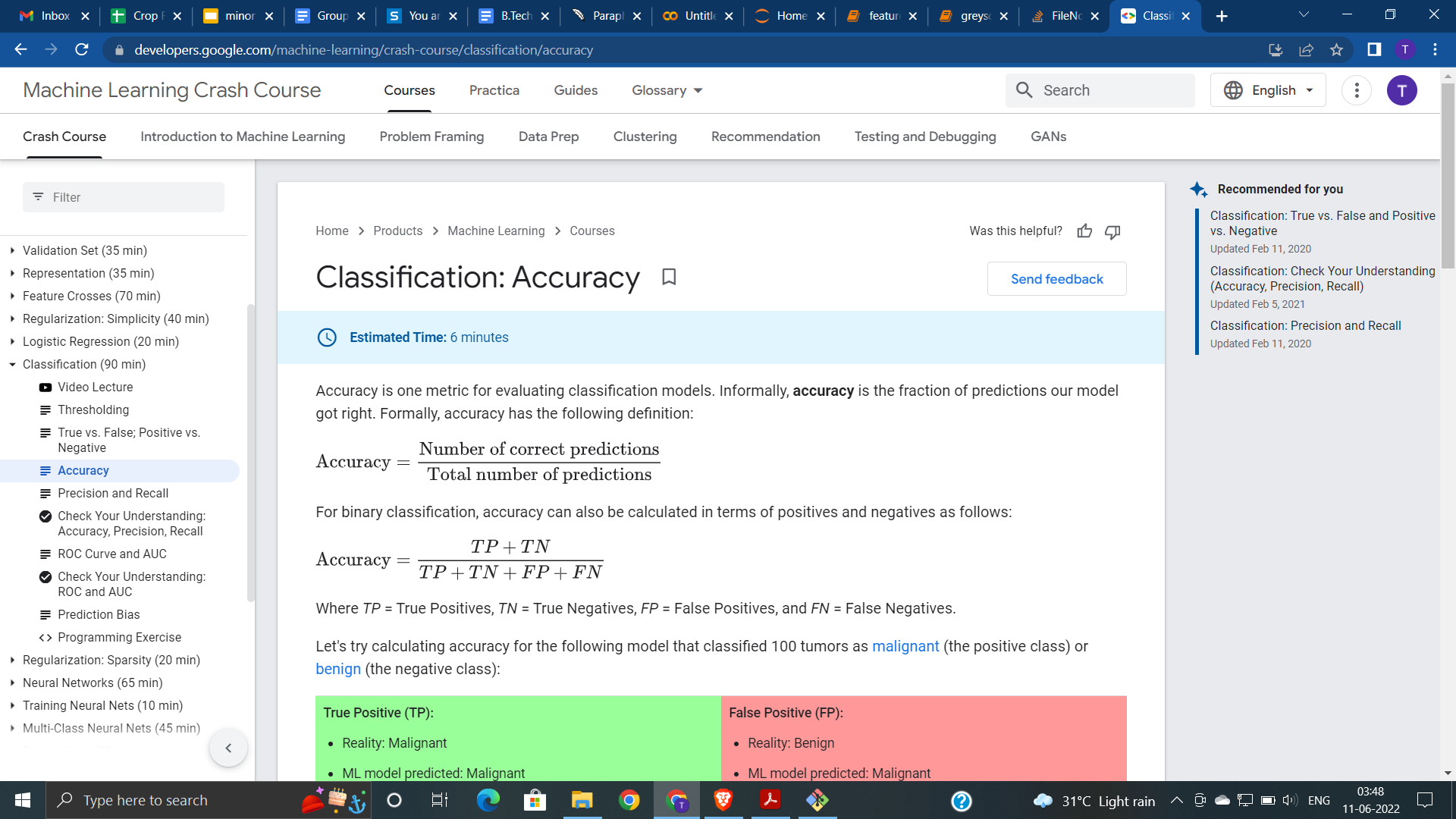
1. Recall =

Recall r is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set ( out of total positive present, how many classified correctly ).



1. F1 Score =

F1-Score is a measure combining both precision and recall. It is the harmonic mean of precision and recall.[17]



1. Accuracy =

*TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

Accuracy is the ratio of correct predictions to total no. of predictions.[18]

**CHAPTER 3. RESULTS**

Visualization of activations in the initial layers of anVGG16 architecture demonstrating that the model has learned to efficiently activate against the diseased spots on the example leaf.

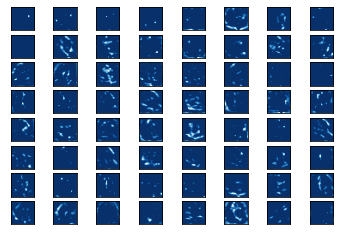
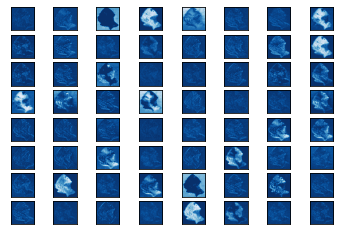


Fig 5. Diseased tomato leaf and its features extracted from 1st and 4th hidden layer

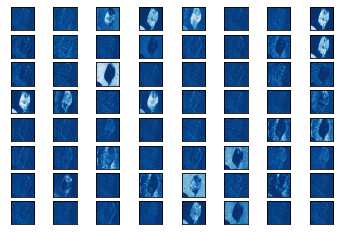
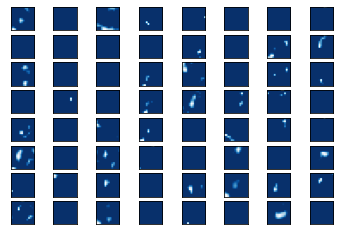


Fig 6. Diseased cauliflower leaves and its features extracted after 1st and last hidden layer

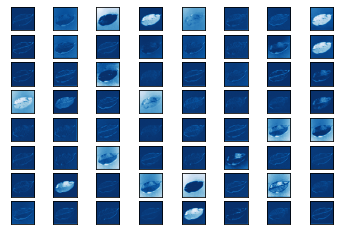
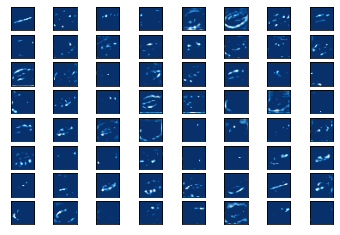




Fig 7. Diseased mango leaves and its features extracted after 1st and 4th hidden layer

**Accuracy analysis:**

Accuracy obtained after training the dataset on all the experimental configurations provided. It can be analysed here that the approx. accuracy percentage of the VGG16 model was 90.795% and the approx. accuracy percentage of the InceptionV3 model was 89.1%.

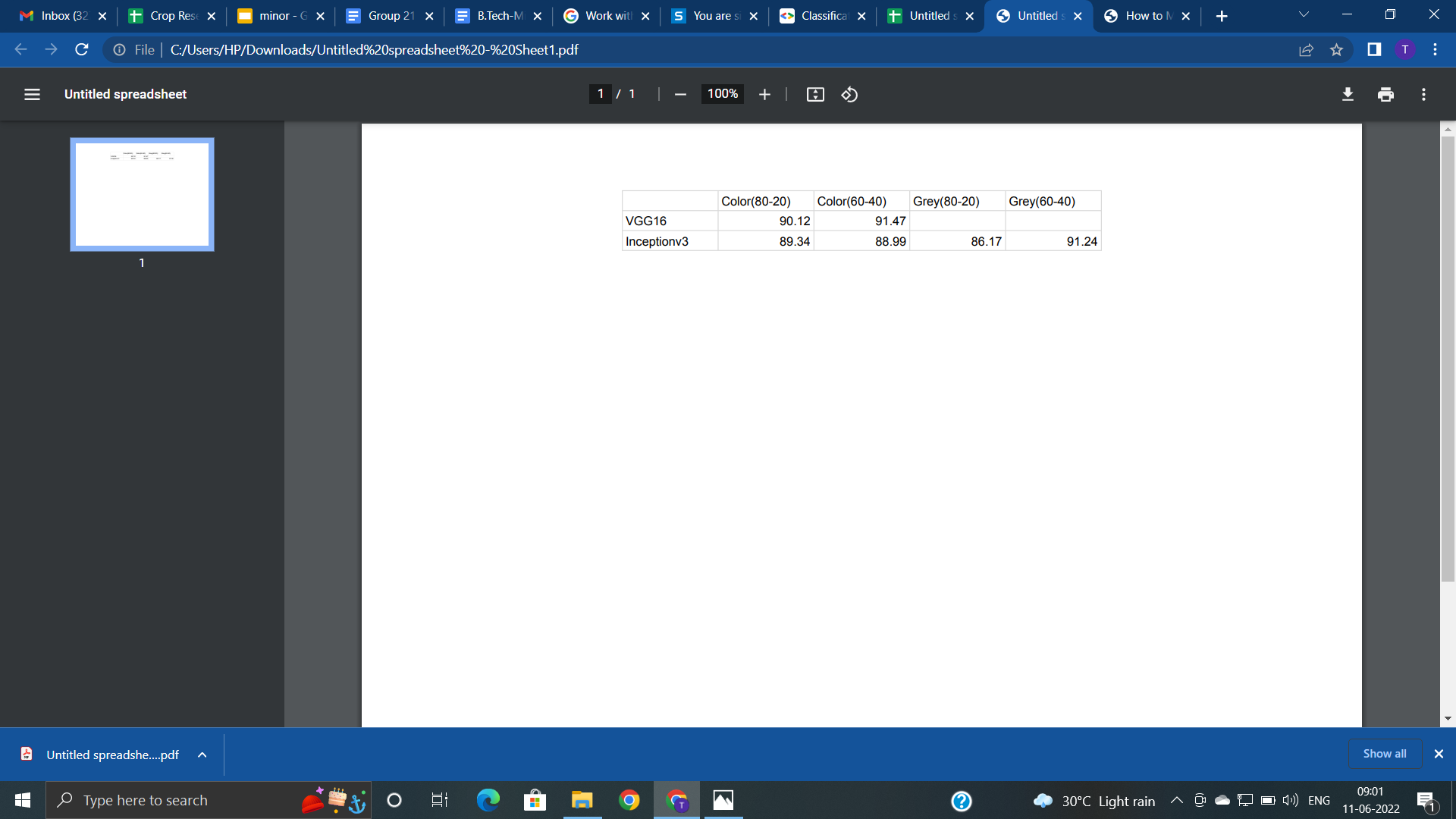


Fig 8. Accuracy results

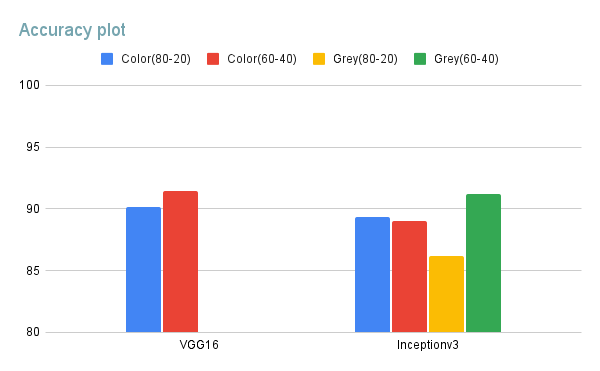


Fig 9. Bar plot of Accuracy

**Evaluation metrics analysis:**

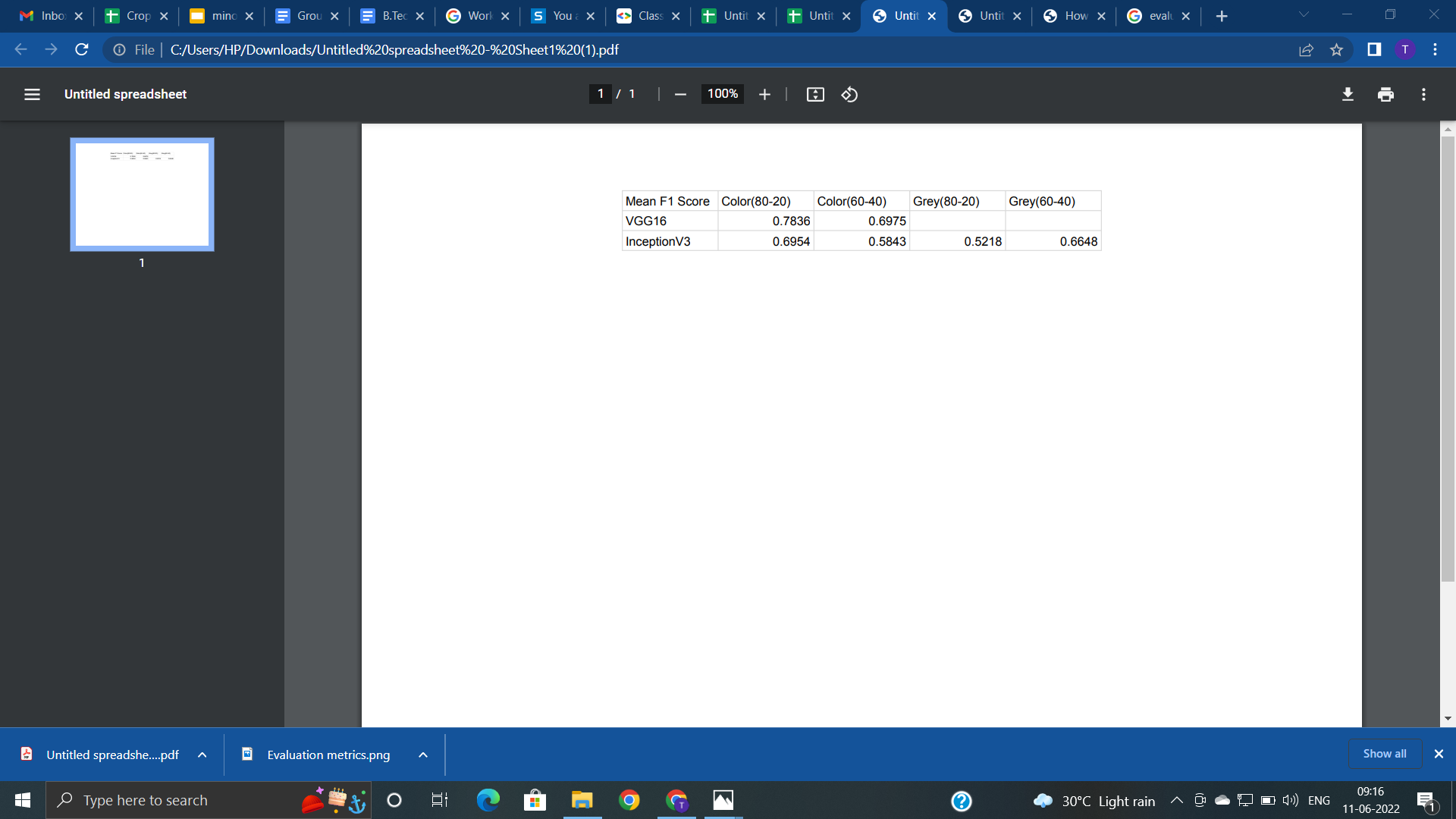
As shown in the figure we can see that the mean F1 scores of all the experimental configurations are mentioned below. The VGG16 model is giving approximate 0.735 mean F1 score and InceptionV3 is giving approximate 0.6137 mean F1 score including all the sub experimental configurations of both the models. Through which we analyse that VGG16 proved to be a better model when considering different evaluation matrices such as recall, precision, f1-score etc. in comparison to InceptionV3 for the dataset used in the experimentation of detection of disease in cauliflower, mango and tomato.

Fig 10. F1 score results

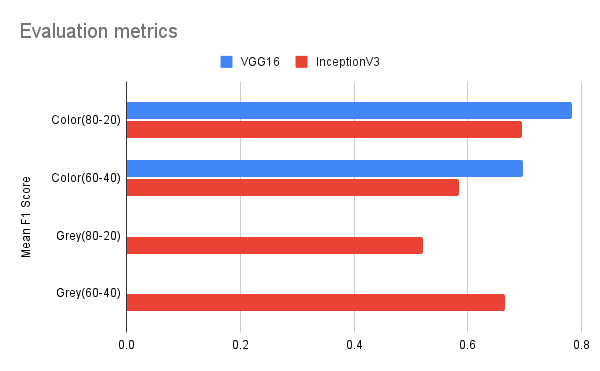


Fig 11. Bar plot of F1 score

**CHAPTER 5. CONCLUSIONS**

Selecting a suitable model for predicting disease affected leaves is important. But the best set of features to predict the diseased leaves can be explored more. In this study we were able to collect our own dataset, and apply VGG-16 and Inception V3 pretrained models to calculate various outcomes. Both the models were applied on coloured and grey-scaled dataset. Also, this dataset was further divided into 2 types of train-test splits, which are 80-20% and 60-40%. For all these dataset splits both the models were applied to predict diseased leaves. the accuracy achieved is as follows :-

Inception V3 - VGG16-

Colour (80-20) - 89.34 Colour (80-20) - 90.12

Colour(60-40) - 88.99 Colour(60-40) - 91.47

Grey Scale(80-20) - 86.17

Grey Scale(60-40) - 91.24

**REFERENCES**

1. <https://ieeexplore.ieee.org/abstract/document/7155951> , Plant Disease Detection Using Image Processing.
2. <https://www.apriorit.com/dev-blog/599-ai-for-image-processing> , Artificial Intelligence for Image Processing: Methods, Techniques and Tools.
3. <https://medium.datadriveninvestor.com/introduction-to-how-cnns-work-77e0e4cde99b> ,Introduction to how CNNs Work.
4. <https://www.kdnuggets.com/2020/02/deep-neural-networks.html> , Deep Neural Networks.
5. <https://arxiv.org/pdf/1409.1556.pdf>,“VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION” by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab of Oxford University, 2014.
6. Mohit Agarwal, Abhishek Singh, Siddhartha Arjaria, Amit Sinha, Suneet Gupta. ToLeD: Tomato Leaf Disease Detection using Convolutional Neural Network. Elsevier 2020.
7. Ananda S. Paymode, Vandana B. Malode. Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. Elsevier, December 2021.
8. <https://ieeexplore.ieee.org/abstract/document/9752096?casa_token=kBWw4fSvwmEAAAAA:kh3HBKMvbFSMow2PxvZuWt7yuHNbNPqEIBOBM4Cd3It4cYDswQHU82bZCAqevoOwHO3Bg9DZg6lCjA> , Brain Tumour Classification using CNN Algorithm
9. <https://ieeexplore.ieee.org/abstract/document/9432235?casa_token=-tWAPQi6ILwAAAAA:DJUFOEQfEIkJlx_LevdbBT6U8wdz04XkavEqY791tYoLpbsUT4ZWcMkLZM0MLJtz6An9v7EV_utH_g> , Brain Tumor Analysis Using Deep Neural Network
10. <https://www.geeksforgeeks.org/vgg-16-cnn-model/> , VGG-16 | CNN model
11. <https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c> , Step by step VGG16 implementation in Keras for beginners
12. ["Rethinking the Inception Architecture for Computer Vision"](https://arxiv.org/abs/1512.00567) by Szegedy, et. al.
13. <https://cloud.google.com/tpu/docs/inception-v3-advanced> , Advanced Guide to Inception v3.
14. <https://sh-tsang.medium.com/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c> , Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015
15. <https://iq.opengenus.org/inception-v3-model-architecture/> , Inception V3 Model Architecture
16. <https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full> , Using Deep Learning for Image-Based Plant Disease Detection
17. [https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3e](https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec), A Look at Precision, Recall, and F1-Score
18. <https://developers.google.com/machine-learning/crash-course/classification/accuracy> ,Classification: Accuracy