Readme

Contents-

1)PROBLEM STATEMENT

The problem statement was that we needed a programme which would analyse the image of the leaf of the maize plant and identify the disease it is suffering from.The input image should have just one leaf of the infected plant.We identified 5-6 diseases seen in maize plants whose symptoms are visible in their leaf and those disease should be detected from the input image provided.

2)METHODOLOGY

There was an ongoing research project taking place in Indian Institute of Technology,Delhi based on the identification of disease from the image of the infected leaf of the maize plant.The project was going to advance in this manner.Firstly, they were going to collect the database from the agriculture department as it was a project given by the government department of agriculture.Then they were going to sort the raw images.In sorting, they had to edit some images and delete some images to get a more processed data to work on.Then they were going to classify the images using transfer learning.They were going to transfer a pre-defined network like Alexnet or Resent18 and then modify some of their layers.Then they would train that network on the data collected by them.For train-

ing they had to manually sort the data into 5 folders where the name of the folder should be the name of the disease suffering by the plant and all images of that disease should be in that particular folder.This is a way to give labels to all the images.Then we would test random images on that trained network and see the result.The training accuracy will be high but the testing accuracy will definitely be low.To improve it we need to do a lot of tasks like image segmentation, object detection and many more.
I contributed to this ongoing project in the following manner.Since the data was not present so I collected data of flowers from net and made a programme which would classify the data into groups.Now as soon as the plant data comes, we can directly run the same programme on the plant data and get the required result.I made a programme which could classify images using Alexnet and another one which would classify images using Resnet18. I made this programme to classify images of flowers into 5 categories of flowers-1)daisy,2)dandelion,3)sunflower,4)roses and 5)tulip.I collected a data of 10,000+ images where they were at least 2,000 images of each flower present.These programmes gave 80-90 percent training accuracy but low testing accuracyThen to improve the testing accuracy I learned image processing in java.In that, I learned image segmentation so that we can give segmented images of the leaf as input for the classification programme to work on.This will improve its accuracy as now only infected leaves of all images go in the input rather than the whole image.I tried various ways to segment out leaves from the image and all went in vain.After continuous efforts, I got a watershed algorithm that could segment out leaves from the image successfully.Now a lot of other things need to be done to improve the testing accuracy to get the required result.

3)IMAGE PROCESSING IN JAVA

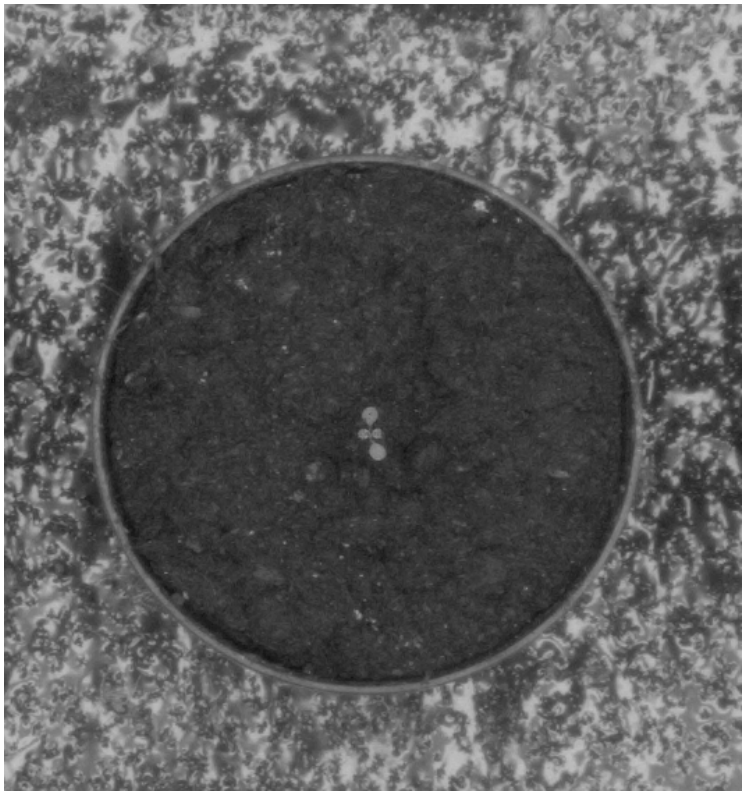
In the beginning, I learned how to read and write an image in java.As I had never used java to modify images before in my life so initially it was a little difficult to understand commands which helps us to read an image file and which help to execute changes on image file.Before this I never knew that I can use java in any other field except for competition programming.The computer reads an images as a matrix or a 2-d array where each elements of the matrix has its own RGB component which indicates the colour of the element.This is how the computer reads an input image.

3.1)GRAYSCALING AN IMAGE-

I wrote my first code which would grayscale the image.The basic concept of the code was to replace the R,G and B value of an element of a matrix with its average value i.e.
(R+G+B)/3.When you repeat this on every element of the image ,it would result in a gray image.

INPUT IMAGE-

OUTPUT IMAGE-

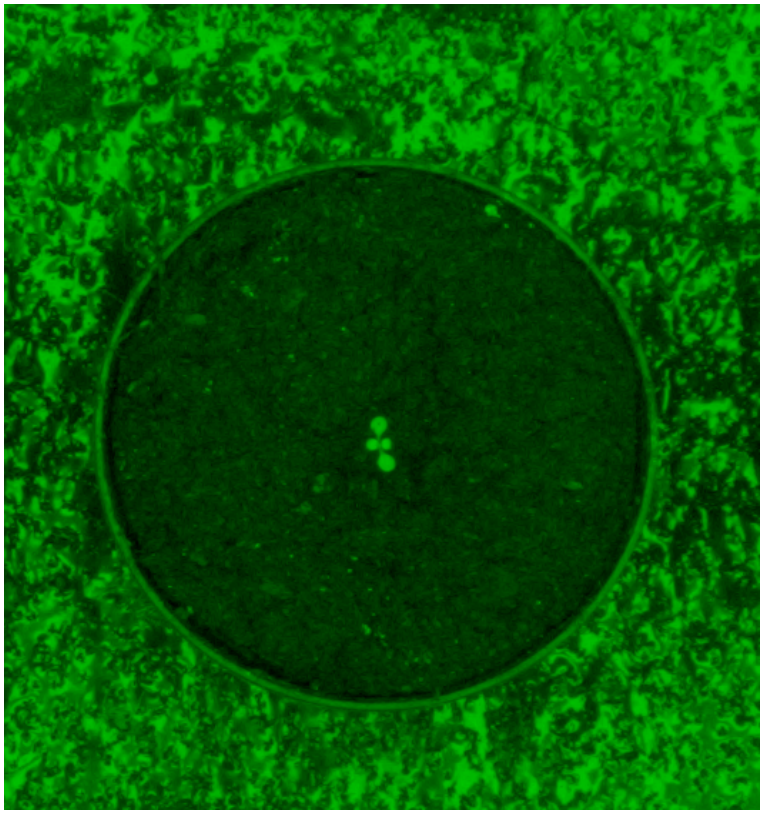


3.2)RED , GREEN OR BLUE SCALING OF THE IMAGE

Initially,I thought that maybe by green scaling the image I might get the required result but this effort went into vain.In green scaling we just retain the green component of the element and set its red and blue component to zero.This happens for all components in the matrix.Similar procedure takes place for red and blue scaling too.I thought that if I retain the green component of each element of the image then only the leaves which have maximum green component will be visible differently and other components which have a minimal green component will be visible differently and so I would be able to differentiate between them and thus segment the leaf out of the image.

INPUT IMAGE-

OUTPUT IMAGE AFTER GREEN SCALING THE INPUT-

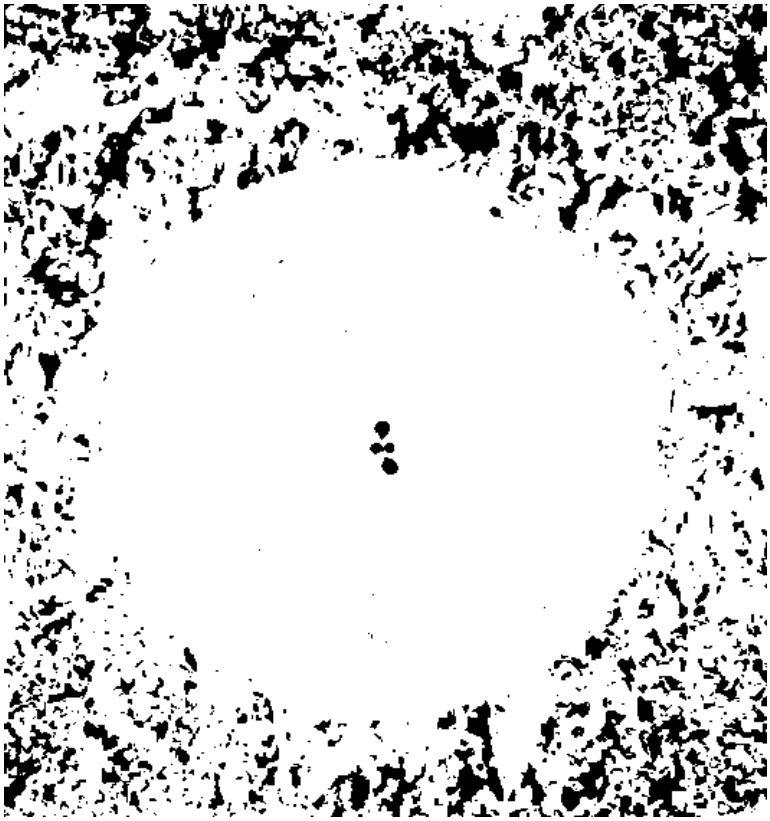


3.3)IMAGE SEGMENTATION USING THRESHOLD METHOD

I thought that if I can convert the image into black and white it would segment out leaves from the image.I decided to make the leaves black and the rest of the background white.So I tried to set a threshold by hit and trial method.I found 135 as the perfect threshold so when the green component of an element is greater than 135 then I set RGB values of that component to 0 each to give it black colour and when green component is less than 135 then I set RGB values of that component to 255 each to give it white colour.This is how all the leaves were turned black and rest of the background was turned white.This segmented the leaf out of the image but the results were not satisfactory.

INPUT IMAGE-

OUTPUT IMAGE-



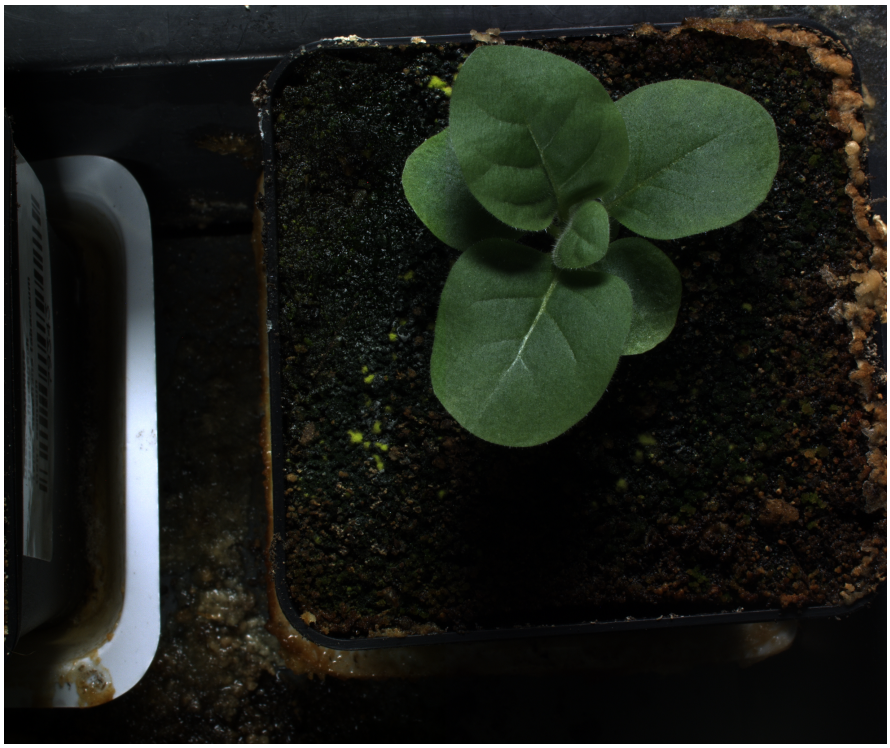
4)FACE DETECTION PROGRAMME

I got to know that I can improve my result using open computer vision.So I started reading about open cv. Initially, I thought lets make a programme on face detection using open cv as that would help me detect a particular object from any input image like face in this case.I thought by this I can also detect leaves on any image and segment it out.Since faces are of numerous kinds and so do leaves so seeing this similarity I started working on face detection.Also face detection will give me a good hold of open cv.For face detection, firstly I needed to know how to find the mirror image to get hold of the position and orientation of the images.

4.1)MIRROR IMAGE

In mirror image we copy the left most component and paste it as the right most component of the new image and so on we keep moving from left to right on the input image and right to left on the output image.So as a result the image gets laterally inverted.Similarly you can also invert an image too.

INPUT IMAGE-



OUTPUT IMAGE-

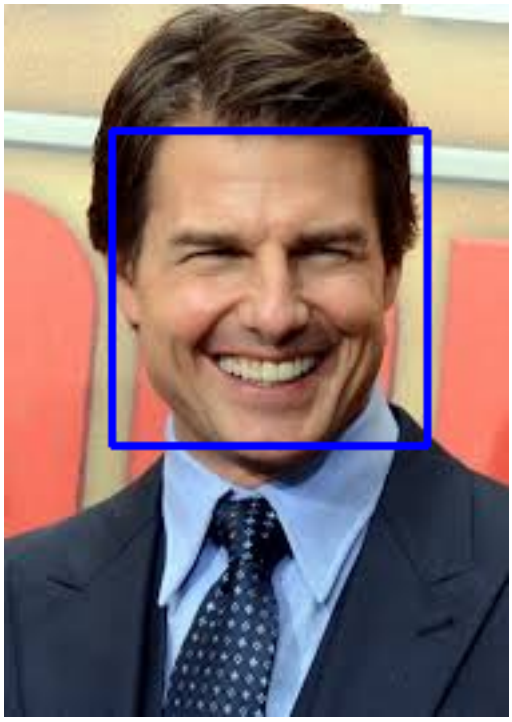
4.2)FACE DETECTION PROGRAMME USING OPEN CV

I took help from a GitHub project for implementing this particular task.There were two xml files one of them had all kinds of faces and other one had all kinds of eyes.These xml files were picked up from Github.The code first reads the input image, converts it into Grayscale and then compares it with the given xml file to locate all the faces and eyes.Then it draws a rectangle around the face as well as eyes.If we want to detect only faces then the code should not read the xml files containing all kinds of eyes.This code below draws a rectangle around a face to identify the area of the face.If we detect the area of the leaves in similar way then we can easily get our results.

RESULT IMAGE-

5)<u>OPEN CV IN JAVA FOR IMAGE SEGMENTATION</u>

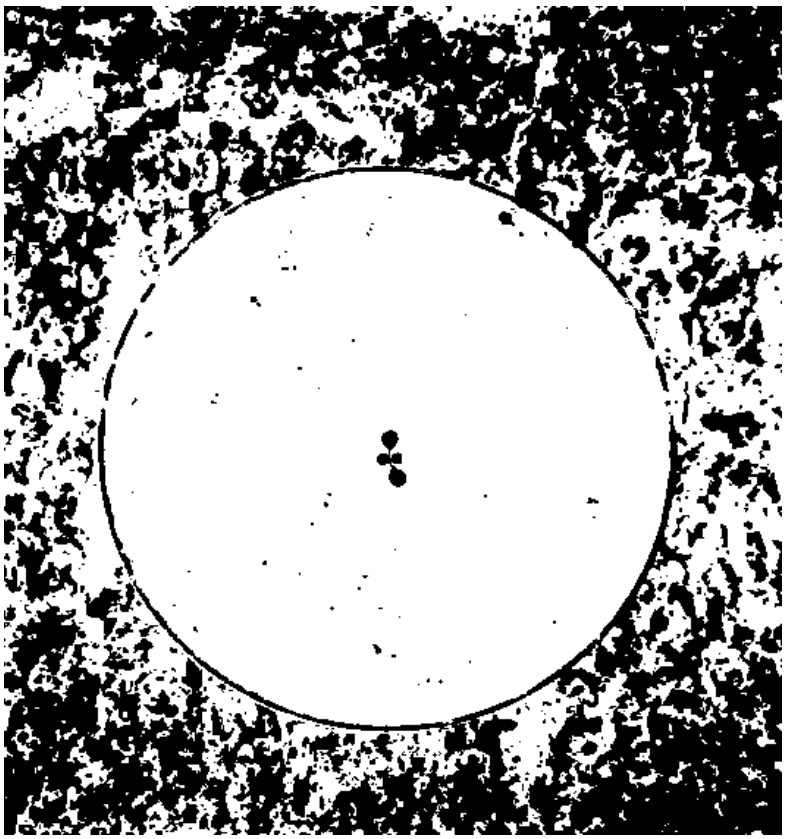
I knew I had to use open cv to get satisfactory results so I kept on trying different ways and then I got one perfect method named **WATERSHED ALGORITHM** which also works by threshold method but in a slightly different manner.
In watershed algorithm a grayscale image is viewed as a surface where high intensity denotes peaks and low density denotes valleys.We start filling valleys with different colour(labels).As water rises, water from different valleys with different colours will start to merge.To avoid this we build barriers until all peaks are under water.These barriers will give segmentation results.



The process which we basically used in the code was **Distance transform along with watershed algorithm** which could segment out objects even if they are connected.In distance transform we need to apply a proper threshold. Next we need to find the area which can be identified as background. For that, we dilate the result. Dilation increases object boundary to background. This way, we can make sure whatever region in background is really a background, since boundary region is removed.In the remaining image ,watershed algorithm will find where they are leaves and where they are not.The border areas are those where objects and background meet and so two colour are meeting.We can also use **EROSION** in place of watershed algorithm if our main focus in the result is only the object and not the background. We could also use a **marker method** in which we give a particular integer to all areas which are identified as objects and we give another number to all background areas and only the unknown areas should be assigned 0.These are the markers. Then we apply watershed algorithm to update the markers with the help of the labels like objects, background and unknown.

INPUT IMAGE

OUTPUT IMAGE-



After repeated failures,I finally figured out how to segment leaves out of the image.This method gave satisfactory results which can be used for further processing.

6)SETTING UP MATLAB ENVIRONMENT

Setting up MATLAB platform in the computer was a challenging task.I installed it as I knew I will need MATLAB platform for image classification.

7)INTRODUCTION TO CONVOLUTIONAL NEURAL NETWORK,DEEP LEARNING AND TRANSFER LEARNING

HOW CNN'S SOLVED THE PURPOSE

I realised after segmenting out images that now my main task is to classify these images under groups.I have total 5000 plus images and I need to classify these 5000 images into 5 groups of almost 1000 images each.We have identified 5 common disease of maize plants which can be indicated by their leaves.We are basically giving labels to all images where labels are the disease a plant is suffering from.When we pass an input image the programme find the image most similar to it and gives the same label to the input image.The programme also devises a group of characteristics from all the images of a particular disease and continues doing it for all the 5 groups.Then it reads the input image and tries to match its characteristics with one group out of 5 which resembles most with the input image and thus gives that particular label to the input image.Thats how classification of

images are done using CNN's.To implement this complicated task ,I need to have a good knack of convolutional neural networks, deep learning and transfer learning.

CONVOLUTIONAL NEURAL NETWORK

A convolutional neural network(CNN) is an algorithm for **Deep Learning**, a type of machine learning in which a model learns to perform classification tasks directly from images, videos, text, or sound.CNNs are particularly useful for finding patterns in images to recognise objects, faces and scenes.A convolutional neural network can have tens or hundreds of layers in which each layer learns to detect different features of an image. Filters are applied to each training image, and the output of each convolved image is used as the input to the next layer. The filters can start as very simple features, such as brightness and edges, and increase in complexity to features that uniquely define the object.

TRANSFER LEARNING USING PRE-TRAINED MODELS

Fine-tuning a pre-trained network with transfer learning is typically much faster and easier than training from scratch. It requires the least amount of data.You start with a pre-trained network and use it to learn a new task. One advantage of transfer learning is that the pre-trained network has already learned a rich set of features. These features can be applied to a wide range of other similar tasks. For example, you can take a network trained on millions of images and retrain it for new object classification using only hundreds of images.Popular pre-trained networks used for classification of images are **Alexnet** and **Res-net18**.

7.1 ALEXNET

It was one of the first deep networks to push ImageNet Classification accuracy by a significant amount in comparison to traditional methodologies. It is composed of 5 convolutional layers followed by 3 fully connected layers.It has ReLU activations attached after every convolutional and fully-connected layer.It consists of convolutions, max pooling, dropout, data augmentation, ReLU activations and many more things.

7.2 RESNET

The architecture is similar to the VGGNet consisting mostly of 3X3 filters. From the VGGNet, shortcut connection is inserted to form a residual network(RES-NET).It uses a global average pooling followed by the classification layer.ResNets were learned with network depth of as large as 152.ResNet-152 achieves top-5 accuracies.

7.3 COMPARING STRUCTURES OF ALEXNET AND RESNET18

ALEXNET

1)AlexNet was designed by the SuperVision group, consisting of Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever.
2)The network has a very similar architecture with LeNet but it is deeper and has more filters per layer, and also it has stacked convolutional layers.
3)It has attached ReLU activations after every convolutional and fully-connected layer.
4)It is composed of 5 convolutional layers followed by 3 fully connected layers.
5)AlexNet has two parallel CNN lines trained on two GPUs with cross-connections.
6) AlexNet was trained for 6 days simultaneously on two Nvidia Geforce GTX 580 GPUs.
7)It reduced the error from 26% to 15.3%.

RESNET

1)It is also called Residual Neural Network (ResNet) and it was designed by Kaiming He et al.
2)The architecture is similar to the VGGNet consisting mostly of 3X3 filters. From the VGGNet, shortcut connection are inserted to form a residual network.
3)It has an architecture with "skip connections" and it features heavy batch normalisation.
4)Each Resnet block is either 2 layer deep (Used in small networks like ResNet 18, 34) or 3 layer deep(Used in ResNet 50, 101, 152).We will use Resnet18 to classify images.
5)ResNet has residual connections.
6) ResNet was made to learn with network depth of as large as 152.
7)It has achieved a top-5 error rate of 3.57%.

HOW DID I ACHIEVE IT?

I tried using Alexnet and Resent on Matlab to classify images.Since the data for the maize plants was not available as it was suppose to come from government agriculture department so I made a programme which could classify images of flowers.The same programme can also be used to classify infected maize leaves into groups of disease.I took images of Roses, sunflowers, Daisy, Tulip and Daffodils from https://garden.org/plants/group/.I made a folder named "flowers" which had 5

sub-folders with the name of the 5 flowers mentioned above.Each sub-folder had more than 2,000 images of that particular flower.
The steps followed for each network is-

1)Load image data.

2)Define the network architecture.

3)Specify training options.

4)Train the network.

5)Predict the labels of new data and compare it with stored data to calculate the classification accuracy

ALEXNET

1)I loaded the folder named "flowers" into Matlab from the desktop and stored it as an ImageDatastore object in matlab. An image datastore allows you to store large image data, and it efficiently reads batches of images during training of a convolutional neural network.Then divide the data into training and validation data sets.Use 70 percent for training and 30 percent for validation.

2)Then I loaded a pre-trained network called Alexnet .I transferred all layers and replaced the last three layers with a fully connected layer, a softmax layer, and a classification output layer. Then I specified the options of the new fully connected layer according to the new data set.Set the fully connected layer to have the same size as the number of classes in the new data.To learn faster in the new layers than in the
Transferred layers, I increased the WeightLearnRateFactor and
BiasLearnRateFactor values of the fully connected layer.

3)The network requires an input images of size 227-227-3.To fulfil that requirement I used an augmented image datastore to automatically resize the training images. Data augmentation helps the network in memorising the exact details of the training images. An epoch is a full training cycle. We can specify the number of epochs according to our need.Then I specified the mini-batch size and validation data. The software validates the network every ValidationFrequency iterations during training.

4)Then I trained the network on the data set of flowers.It will take as many epochs as I mentioned while training the data.

5)Then I classified the validation images using the fine-tuned network called "Alexnet". Then I calculated the classification accuracy on the validation data set by comparing the predicted result with the stored result for each image. Accuracy is the fraction of labels that the network predicts correctly.

RESNET

1)Load data into imagedatastore and divide it into training and validation sets.

2)Load the pretrained ResNet-18 network.To retrain ResNet-18 to classify new images, replace the last fully connected layer and the final classification layer of the network.Set the new fully connected layer to have the same size as the number of classes in the new data set. To learn faster in the new layers than in the transferred layers, increase the learning rate factors of the fully connected layer.

3)The network requires input images of size 224-224-3, but the images in the image datastore have different sizes. Use an augmented image datastore to automatically resize the training images.Specify the training options, including mini-batch size and validation data.We set the learning rate settings such that there is faster learning in the new layers and slower learning in the other layers.

4)Now trains the modified network on the data set of the flowers.

5)Classify the validation images using the fine-tuned network, and calculate the classification accuracy.

HOW TO TEST IMAGES-

In both networks after training is done we use the same procedure to test the classification of images.We collect the images whose label we want to know in a folder.We first load those image into Matlab. We resize it according to the network using imresize() command.Then we classify the images using the newly trained network which was trained on the data set collected by us.This classification yield a label to the image.We print the label in Matlab and hence get the desired result.We can also get the picture of the image in matlab using imshow() command.

CODES MADE DURING THIS INTERNSHIP-

All the codes for image segmentation, image processing and the Matlab code for image classification are given in the link below.This link has all the codes I made during the course of this internship.

https://drive.google.com/drive/folders/1Tcir48d-mKTuTC2Cy5kYi_TuRXCYMGYi?usp=sharing

8)CONCLUSION

This internship gave me an opportunity to try out new things apart from my curriculum.Before this internship I knew nothing on how to even read or modify any image but by the end I could modify the images and also classify them based on some prominent features.I could now play with images, invert them, grayscale them and take their mirror image.I also got to know about different methods of segmenting images like threshold method and watershed algorithm.I could now successfully segment out images using watershed algorithm.I could now implement things like face detection using open computer vision which gave me a knack of object detection. This internship taught me about different pre-trained networks and how you can transfer them and modify them according to your own need.It also taught me how to train your own data on the transferred network and then test it.It taught me all about convolutional networks, deep learning and transfer learning.It taught me the structure and functioning of two popular networks called Alexnet and Resent18.To sum it up, this internship gave me a knack of image processing, OpenCV, transfer learning, deep learning, Alexnet, Resnet18 and convolutional neural networks