Term Project Final Report

CSE 583: PATERN RECOGNITION, Spring 2020

Umar Farooq Mohammad

May 5th, 2020

Project Title

Plant leaf disease detection and classification

Problem Statement and Motivation

In this project, I intend to build classification model for the detection of the type of plant and its disease form the image of the diseased leaf. In simple words if we give the below image as an input without any further information. My model will have to detect it as "**A Grape leaf with Black-Rot Disease**"

Figure 1 Sample of diseased leaf

Why I have selected this problem statement for my term project?

We all know agriculture is of very high importance for humans to survive. In recent days we are seeing many different types of virus and bacteria effecting the Plants and humans as well. Being from a country India where agriculture is the major occupation across the country. I want to use my knowledge and expertise for implement something useful and beneficial to over come the problems which are faced regularly in agriculture.

Early detecting of the plant disease before it destroys the crop will be of major advantage to prevent the complete crop from being destroyed and also to get a high yield from the crop, so by using this image classification techniques and latest advancement methods of deep learning we can easily detect the correct disease of the plant and so that we will be providing the right pesticide to overcome the disease for plant.

Farming experts can directly identify them visually but we cannot completely reply on the experts for all the crops across the globe, its not an easy task to monitor all the crops by the experts, Due to recent advancements in the technology if we are able to introduce this model in the form of an mobile application then any farmer can easily detect the plant disease with just a photo of the leaf from their smartphone. And one more advantage is that we can also collect these pictures as a dataset for the future research to detect any new type of diseases.

So, introducing the artificial intelligence into the detecting of these plant diseases is of high advantage and necessary.

Literature Survey

There are a bunch of research works done in the area of detecting the plant disease using the leaf image. Most of the papers were focused on only 1 specific crop like tomato or grape plants etc. There is no research done for the classification of diseases across different types of plants. So my I am trying to implement the classification across different types of plants and their diseases using the Deep learning techniques.

In [1] the authors have worked on the classification of the diseases in papaya leaves, They have used the Histogram of oriented gradient(HOG) method for the feature extraction and as a preprocessing they have converted the RGB image into Gray scale to suppress the domination of the back ground and also to extract the Hu moments features from the image. For the training of model, they have used the random forest algorithm on the extracted feature vector. In my work I would like to use CNN techniques as it will directly look out for the best performing features from the image for training the model. The accuracy of 70.14% was obtained using the random forest algorithm.

In [2] the authors have used the CNN method on the dataset of 500 tomato leaves images, the authors have achieved an average accuracy of 86% on their dataset. They have used Learning vector quantization (LVQ) as the classifier. The have fed the output form the training network as an input to the LVQ classifier algorithm.

In [4] the authors have used the image processing techniques to extract the best performing features from the diseased leaf for the learning model. They have converted the RGB images into L*a*b color space for the enhancement of visual analysis. K-means clustering was used for the image segmentation for the best detecting of the infected areas in the leaf.

For the feature extraction part GLCM (Gray level co-occurrence matrix) is used an SVM method is used for the classification of the images to their disease type. The GLCM functions characterize the texture of images by computing the spatial relationship among the pixels in the images.

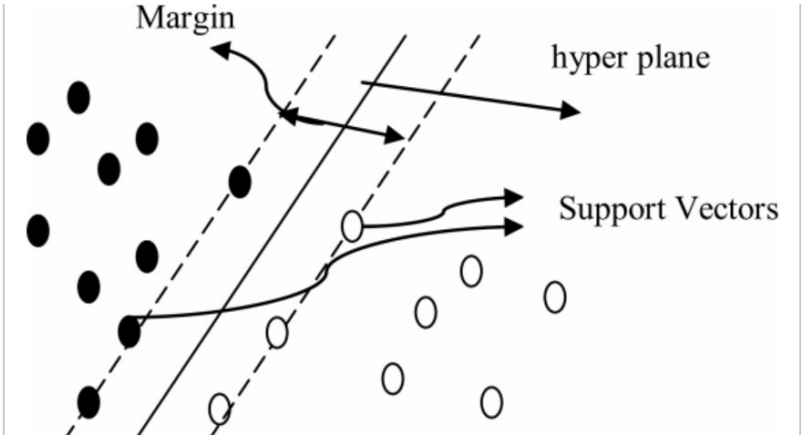


Figure 2 GLCM Feature extraction representation

They have achieved an accuracy of around 90%. This work was done only on the dataset of citrus leaves.

[5] This is one of my interested paper for the preprocessing task done by the authors. They have performed image acquisition for the enhancement of the images to remove the noise added to the images by the digital cameras while they were captured. So, by clipping, smoothing & enhancement using the median filter they have got the enhanced image without noise. And as a next step in preprocessing of data they have done segmentation which is important for extracting the image from the complex back rounds which is common for the pictures of leaves. The training and classification in this paper was done in the way like [4] & [5] using the Feature extraction and SVM classification techniques.

The work done in [6] is like the work of [2] but the authors of [6] have done some extra preprocessing of the data such as scaling and augmentation. As the pictures in the dataset are the clear images of only leaf so scaling will not affect the originality of the image but as we are trying to reduce the pixels directly there is a risk of losing the diseased spots on the leaves.

The paper [7] is a nice comparison research work done to test the accuracies of the leaf disease detection done using the already pre trained networks AlexNet & SqueezeNet and the author have got an accuracy of 95% for AlexNet and 94% for SqueezeNet but the Inference time of the SqueezeNet was very less than that of the AlexNet, Its almost 1/3rd of the time taken for AlexNet.

Dataset Description

The dataset I am using for this project is "Plant Village" Dataset which contains the cropped images of the leaves of different plants which are of 256*256 pixels size in all the images.

There are 38 different categories of data in total dataset which are of different types of plants with different diseases. The diseases of all the plants are not the same, each plant has its own different diseases based on its type. i.e tomato has 10 different categories in the dataset where 9 different diseases images and the other is the images of healthy images of tomato leaf.

While Potato has only 3 categories in which 2 different diseases and healthy leaf of potato images are provided.

Most of the images provided in this dataset are correctly cropped for the leaf with simple back round so it is beneficial for out training model to extract the correct diseases part of the leaf for the best performing features.

The below is the list of the image categories available in the dataset I am going to use for the project.

Apple:

a) Scab
b) Black rot
c) Cedar apple rot
d) Healthy

Blueberry

a) Healthy

Cherry

a) Healthy
b) Powdery mildew

Corn

a) Cercospora Leaf Spot
b) Common Rust

c) Healthy
d) Northern leaf blight

Grape

a) Esca (Black Measles)
b) Healthy
c) Black Rot
d) Isariopsis leaf spot

Orange

a) HaunglongBling

Peach

a) Bacterial Spot
b) Healthy

Pepper

a) Bacterial Spot
b) Healthy

Potato

a) Early Blight
b) Healthy
c) Late Blight

Raspberry

a) Healthy

Soybean

a) Healthy

Squash

a) Powdery_mildew

Strawberry

a) Healthy
b) Leaf scorch

Tomato

a) Bacterial Spot
b) Early Blight
c) Healthy

d) Late Blight
e) Leaf Mold
f) Septoria Leaf spot
g) Spider mites Two Spotted
h) Target Spot
i) Mosaic Virus
j) Yellow leaf curl Virus

The above 38 are all the different types of categories of leaves in the dataset which I am going to classify by the model.

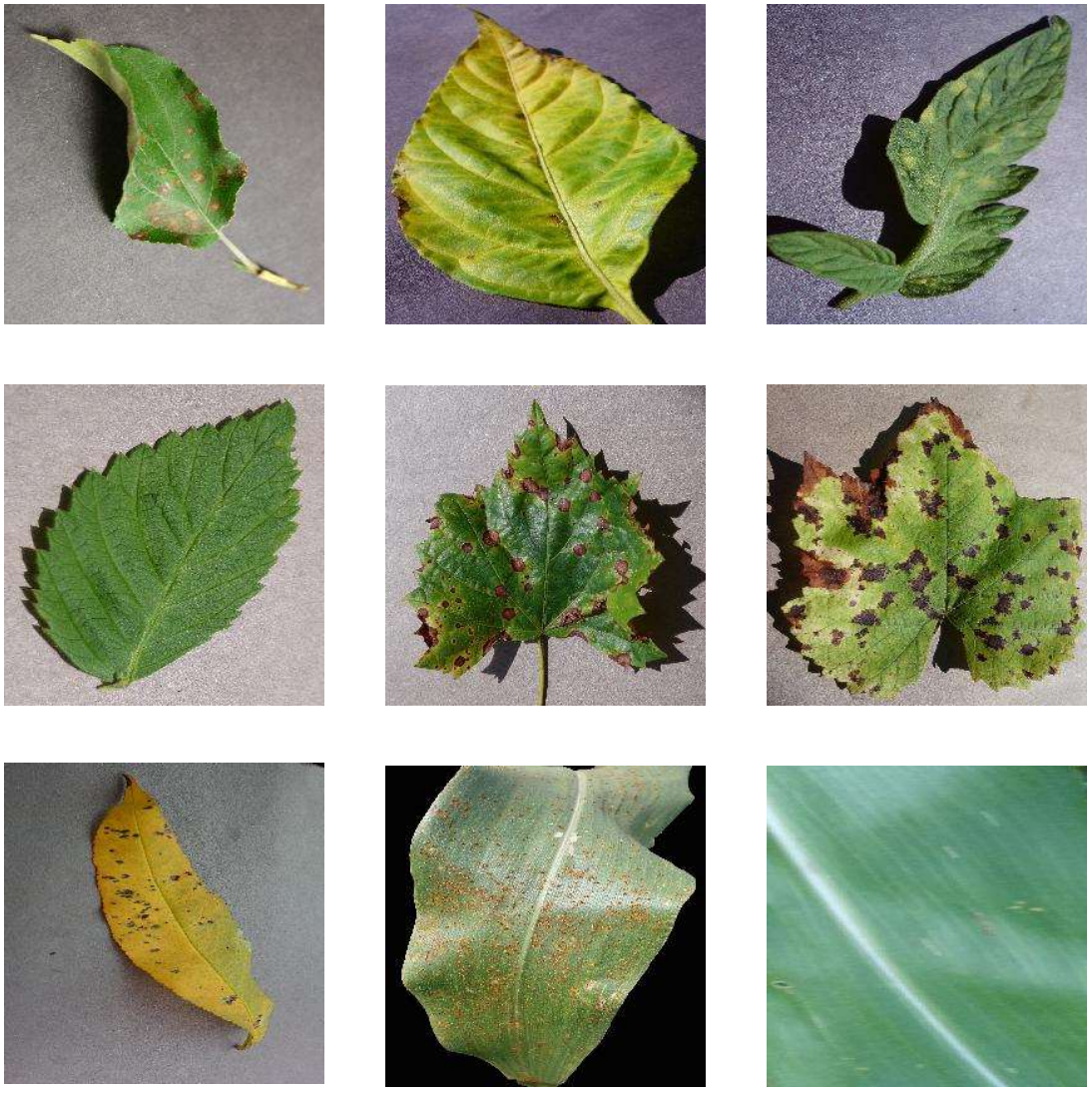


Figure 3 Sample images from the dataset

My dataset contains almost 2000 images for each of the category by making it a total of 80k images for all the leaves used for this classification.

I would like to divide the data into the following proportion for train, validation and test.

Dataset Distribution

Training	Validation	Test
70%	15%	15%

From the above percentages we are getting almost 50k images for all the classes in total. As the images are of the resolution 256*256 pixels color images. From the cited papers which have used just a small part of the dataset I am using they have achieved an acceptable accuracy in the classification. So, I can say that the data is sufficient enough for this 32-class classification problem. But there might be an issue of over fitting as we run for multiple number of epochs and the model may over fit. So, to avoid this issue I am going to use the data augmentation for the better generalization of the classes of the leaf diseases rather than over fitting to recognize specific images.

Link to repository: https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw/color

From the above repository I have downloaded the raw dataset and worked using that data for the rest of the project.

Proposed implementation

I would like to implement the classification for the entire dataset, as all the reference papers I mentioned in the literature survey have only worked on a specific type of plant to classify their diseases. But unlike that I would like to classify the type of plant and its disease using my model, i.e. I would like to build a classification of complete 38 classes in my model.

Pre-Processing:

Segmentation

As the dataset I am dealing with is of leaves images there is a possibility of the loss in accuracy due to the background colors from that of the leaf. So in order to overcome this issue I would like to perform the segmentation for all the images before feeding them to the CNN network for training and extracting the best features.

So that the extracted segment will not have the background other than the leaf, so in future I we want to test using an image with any complex background also my model will not fail to classify as I am using the segmentation of the data to separate the background.

Augmentation:

While we try to train the model for more epochs, then the model will slowly start to over fit, learning to recognize the specific images in the training set, rather than generalizing, such that I also get good results on the validation set. One way to fix this is to effectively create more data, through data augmentation. The below is an image of an unhealthy leaf from my dataset.

Figure 4 Unhealthy leaf before augmentation

The below is the image of the leaf with different augmentation techniques such as horizontal flipping, Zooming and rotating.

Figure 5 The images of the leaf pixels rotated left, rotated right, translated in x,y direction respectively.

I would also like to experiment in pre-processing step by changing the RGB image data into gray scale as it is very easy to detect the image outline of the leaf in gray scale image than the RGB as done in [1].

Training:

In this project am going to make use of the transfer learning as in [7] & [8], I the paper [7] the authors have compared the accuracies with AlexNet and SqueezeNet. I would like to experiment the transfer learning with the AlexNet (SGD), AlexNet(Adam), ResNet & GoogLeNet(SGD) for comparing the their performances and get the best performing model from them.

The size of input requested by AlexNet is 227*227 pixel so while using this transfer learning model I must convert the images into the specified size and while the GoogLeNet accepts the size of 224*224 pixels. ResNet is same with 224*224-pixel size.

I will experiment between the different activation filters and play with the hyper parameters of the model in order achieve the best accuracy. So, I cannot finalize on this part at this part of time for the proposal.

In most of the reference ReLu was used for the activation filter so, I will start experimenting using this filter at first and try out different combinations for the depth of the filter layers for the convolution step.

As my model must classify the 38 different classes I will modify the last step i.e. FC network to make it compatible to classify the 38 classes

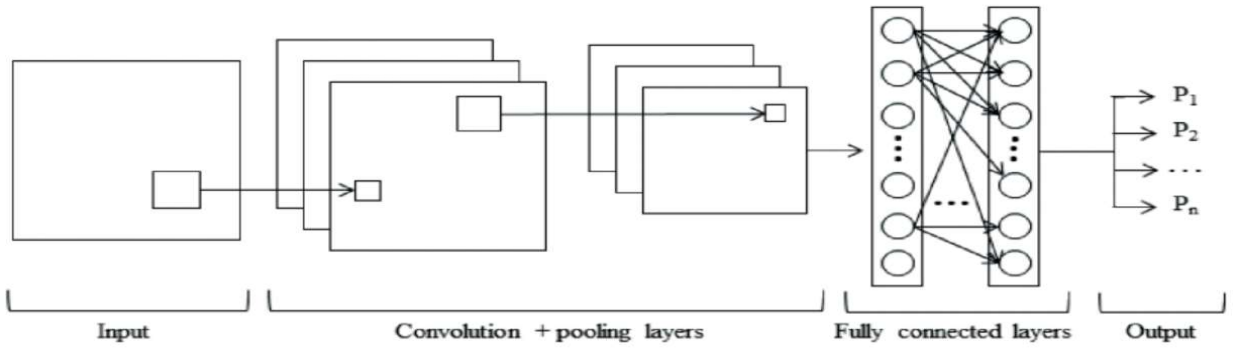


Figure 6 Reference diagram of CNN

Goals

- To perform the perfect data pre-processing which suits for my classification problem perfectly.
- Build the training models using the transfer learning from the pre trained network architectures available online and compare their performances for my problem statement.
- Achieve an acceptable accuracy for my classification problem.
- Experiment with different hyper parameters, Data preprocessing steps and understand their effects on the performance of the model for the classification of multi class problem.

Alternative Design

As I am using the CNN for the first time for this project, I would like to explore most of the possibilities and features available for classification using CNN, As feature selection in CNN will learn itself for the best performing model from the training data.

If I am unable to achieve the acceptable accuracy by my design as mentioned I will perform the preprocessing of data for the best performance and try to achieve the best accuracy for my model. Even though if at all the deep learning procedure doesn't work out for this problem then as the work done in [4], [5] & [6] I will try to implement classification model by using the feature selection from HOG or GLCM method and then SVM method for the classification.

Tentative Proposed Timetable

Tasks	Deadline
Literature survey and Understanding complete dataset	24th March
Experimenting on the dataset with simple classifications & Learning how to run code using aci GPU's	4th April
Building the proposed architecture	13th April
Training the model & Fixing problems faced or Design changes.	22nd April
Experiment on the model for the different scenarios of expected results.	25th April
Final Results and Documentation	28th April

Implementation

The below sections are the information about my implementation for this project, Which includes all my experiments with the dataset to get good accuracy with initial and final results. The visualizations and the observations from them are also included below

Data-Pre-Processing:

As mentioned in the proposal the data is augmented before the training of the dataset for the desired model and the initial results were reported in the below sections. As the desired results were not achieved with the raw and augmented data some more additional pre-processing techniques were used for the better enhancement of the train image so that the network will easily detect the best performing features from the leaf image for detection.

Gray scale & edge detection:

As the leaf type detection is mostly dependent on the shape of the leaf as that will differ in between various types of leaves. So as per the paper [3] the similar approach is followed to covert all the train data to gray scale and in the network before testing the test image is also converted to gray scale before feeding to network for prediction.

Figure 7 Raw Image

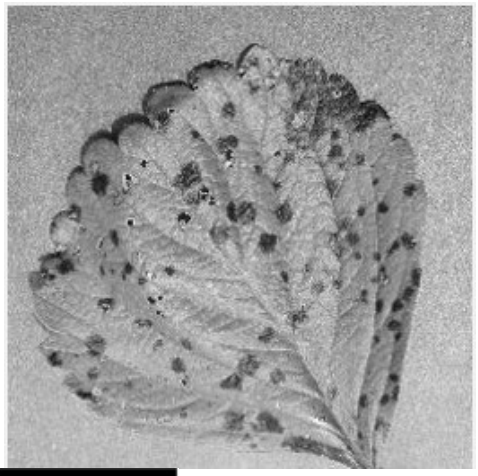


Figure 8 Gray scale image

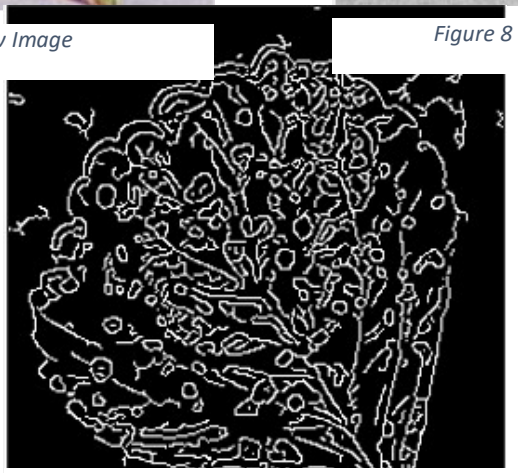


Figure 9 Edge detection

The results with this pre-processing step will be discussed below in the initial results section.

As we can see that in the edge detected image the features of leaf disease spots are not retained and only the edges are present in the image. As there is no meaningful data about the dark spots are retained this type of pre-processing will decrease the performance of the network.

Our main aim is to enhance the images in this step such that the edges will be detectable easily from the background to fore ground and the leaf spots and wrinkles should be detected better. In order to achieve this, I have adapted some of the methods from [5].

Where the image will be enhanced such that to segment the leaf from the background using the segmentation technique “Thresholding”.

Image thresholding is way of converting a color image into a binary image based on certain threshold of pixel intensity. This is very useful in extracting dominant foreground and background objects. It can also be used to create a sketch like images. Here I have used the multi-level thresholding using the Otsu's method for the RGB images.

We have a choice of select the number of threshold levels in this step for the RGB images, I have experimented between different levels of thresholding in order to see which is the best fit to get the better suppression of the background from the leaf image and also along side the leaf disease spots to be enhanced for better feature detections.

Figure 10 Raw Image

Figure 12 Two-Level Thresholding

Figure 11 Single-Level Thresholding

From the above images we can observe that the thresholding has very much enhanced the image. As compared to two level thresholding in the single level thresholding the image background is being suppressed almost and those pixels were converted to white.

When comparing with the raw image which contains some high intensity pixels even in the background so there is a risk of some features being detected and trained by the network while we train with the raw data. Thus, by using these threshold images we can overcome this risk of unwanted features and only train the model on the segmented image of only leaf. And also in the images the easy difference between the spots and the green part of the leaf are better in this image as the color distribution is suppressed and the low intensity pixels are being masked by the thresholding and pixels with intensity higher than threshold will be highlighted.

So, these segmented images should improve the performance of the network training and we will observe them in the experiment results in the sections below.

Experimental Implementation:

At the start of the project I have followed the paper [1] as they have worked with the HOG method to extract the features form the image and experimented with different classifiers for the classification for only 6 classes of the dataset i.e. a subset of this dataset. So, I have followed the same procedure and implemented the classifier using these HOG features. But I was not able to get high accuracies with these classes. As I understood that the classification of these leaves majorly depends on the shape of the leaf. But the HOG features were not being able to provide any meaningful about the shape of the leaf as they depend on the color intensity difference only majorly. As we can see in the below pictures that the top performing features were majorly the green pixels, dark spots and leaf boarders only.

Figure 14 Top 25 features

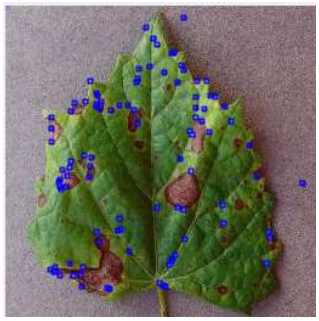


Figure 13 Top 50 features

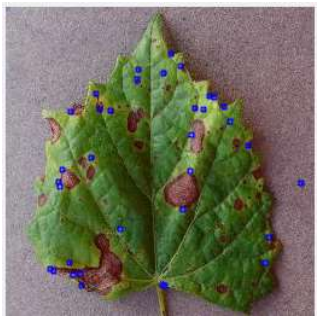


Figure 15 Top 25 features

For Binary classification – 72.8% (Diseased vs Healthy) 2 classes

For Disease Classification – 34.2% for 38 classes

So, these features have the most information about whether the leaf is diseased or not. So I have experimented for the binary classification using these features to detect whether the leaf is diseased or not only i.e. binary classification then I was able to get the reasonable accuracy but for the 38 class classification I was not able to get acceptable accuracies for this dataset classification. Thus from these observations I have concluded that this approach will not be perfect for this multi class problem as I have referred to [11] where they have used deep learning approaches for this problem on the same dataset and the authors were able to get the acceptable accuracies. So, I have proceeded with this approach and below are the results for that.

Initial Training Results:

As the first step of the training process I have trained the model using Transfer learning of AlexNet with the augmentation on the raw data without any pre-processing for the 6 Epochs.

The below are the results of this step.

Batch Size: 350

No of Epochs: 6

Input Image Size: 256*256*3 (Converted to 227*227*3 before training & testing)

Pre-Processing Steps: Augmentation (scaling, translation, rotation)

Learning Rate: 1E-5

Architecture: AlexNet

Data Split: 70%: 15% : 15%

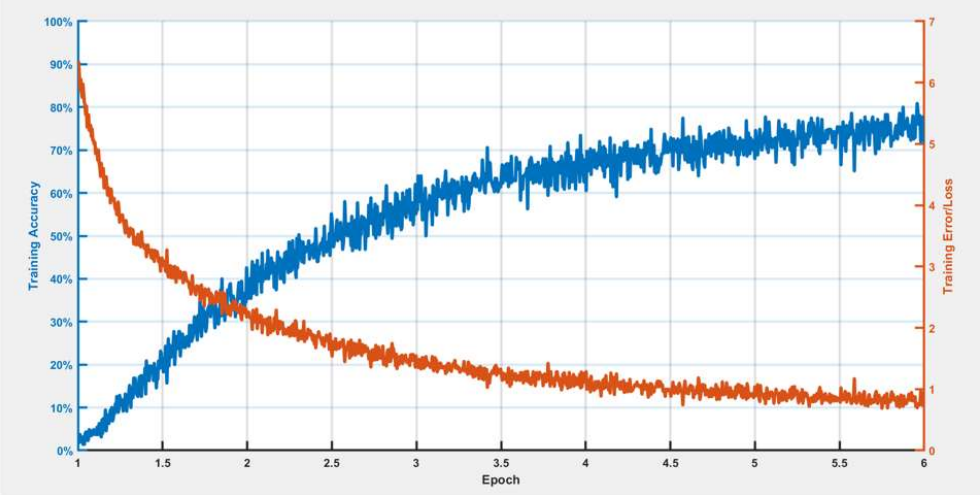


Figure 16 Training rate and loss plot for transfer learning using AlexNet

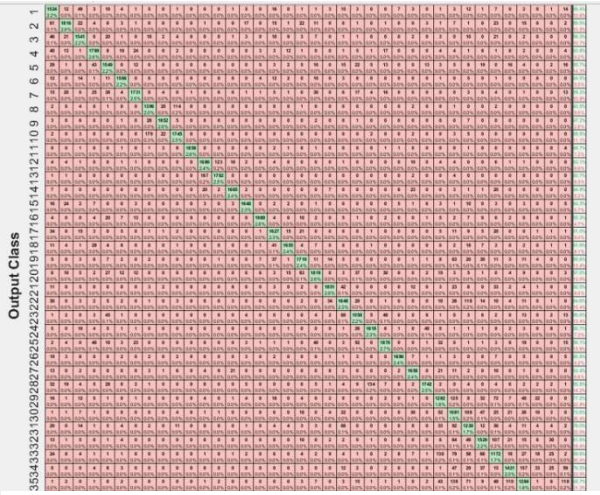


Figure 18 Confusion matrix on training data

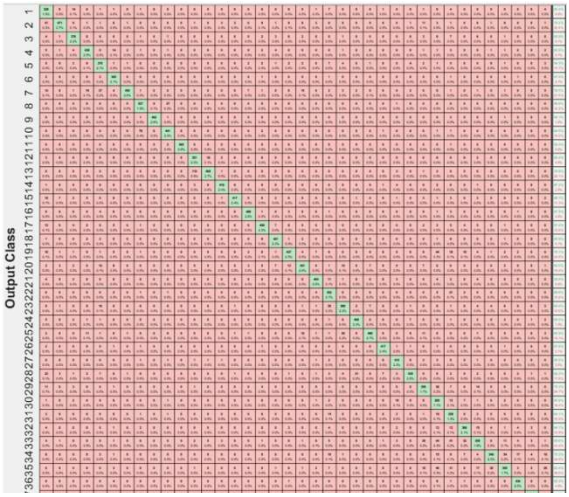


Figure 17 Confusion matrix on validation data

Figure 20 Visualization of 4th convolution layer in network

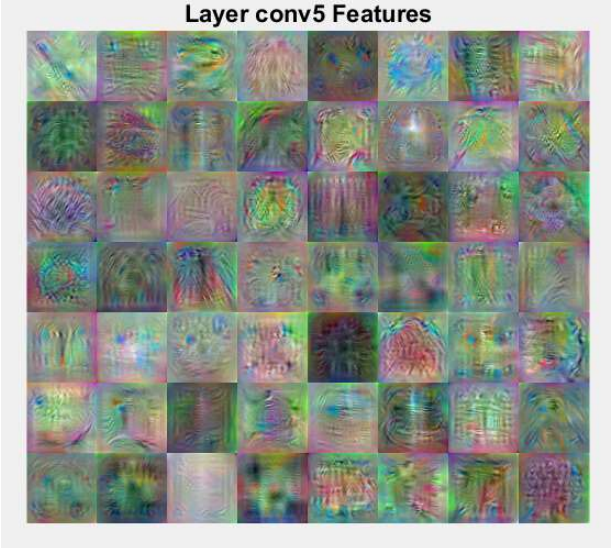


Figure 19 Visualization of 5th convolution layer in network

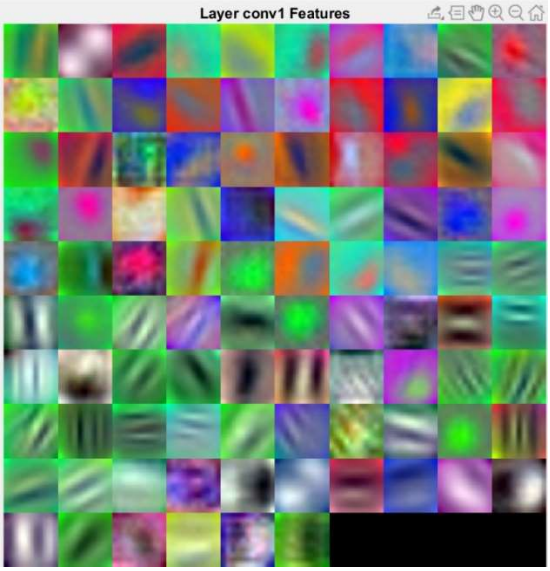


Figure 22 Visualization of first convolution layer features

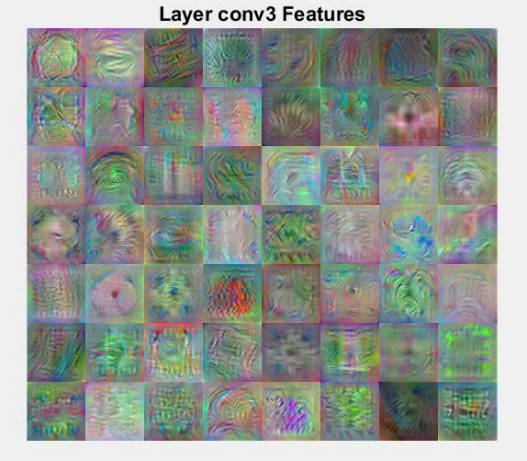


Figure 21 Visualization of 3rd convolution layer of the network

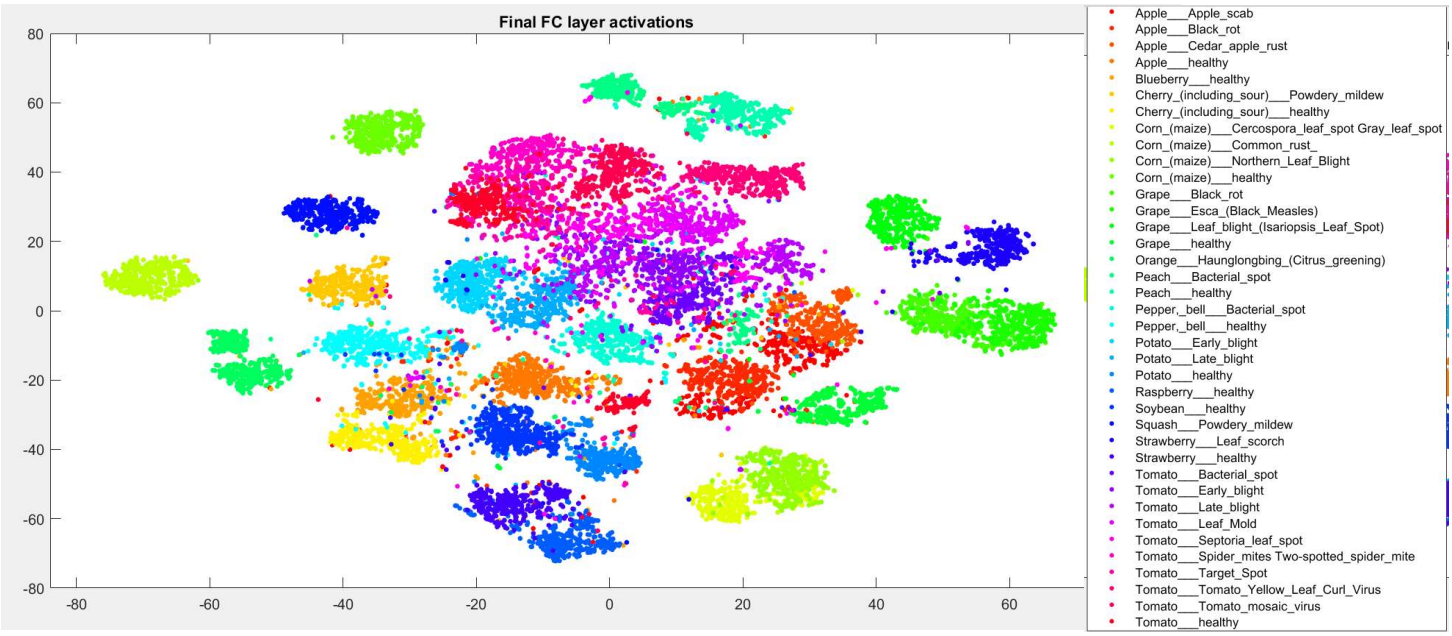


Figure 23 t-SNE visualization of FC layer with AlexNet transfer learning

Network Architecture:

Layer	***Size***	***Bias***	***Activation***	***Kernal size***	***Stride***	***Parameters***
Input	*227*227*3*	..	..	..	..	
Convolution	*55*55*96*	*96*	*relu*	*11*11*	*4*	*34944*
Max pool	*27*27*96*	*96*	*relu*	*3*3*	*2*	
Convolution	*27*27*256*	*256*	*relu*	*5*5*	*1*	*307456*
Max pool	*13*13*256*	*256*	*relu*	*3*3*	*2*	
Convolution	*13*13*384*	*384*	*relu*	*3*3*	*1*	*885120*
Convolution	*13*13*384*	*384*	*relu*	*3*3*	*1*	*663936*
Convolution	*13*13*156*	*256*	*relu*	*3*3*	*1*	*442624*
Max pool	*6*6*256*	*256*	*relu*	*3*3*	*2*	
Fully connected	*4096*	..	*relu*	..	..	*37752832*
Fully connected	*4096*	..	*relu*	..	..	*16781312*
Fully connected	*38*	..	*Softmax*	..	..	*155686*
Output classification	..	..	..	..	..	

From the above results where the accuracy is around 85%, we can say that there is a lot of scope for improvement and increase the accuracy. The state-of-art for this dataset classification have achieved an accuracy of 90% using AlexNet and 99% for the google net architecture [11].

Now the below are the present status of the network accuracies with the pre-processed data where I was able to get and accuracy of around 95% with the GoogleNet Architecture.

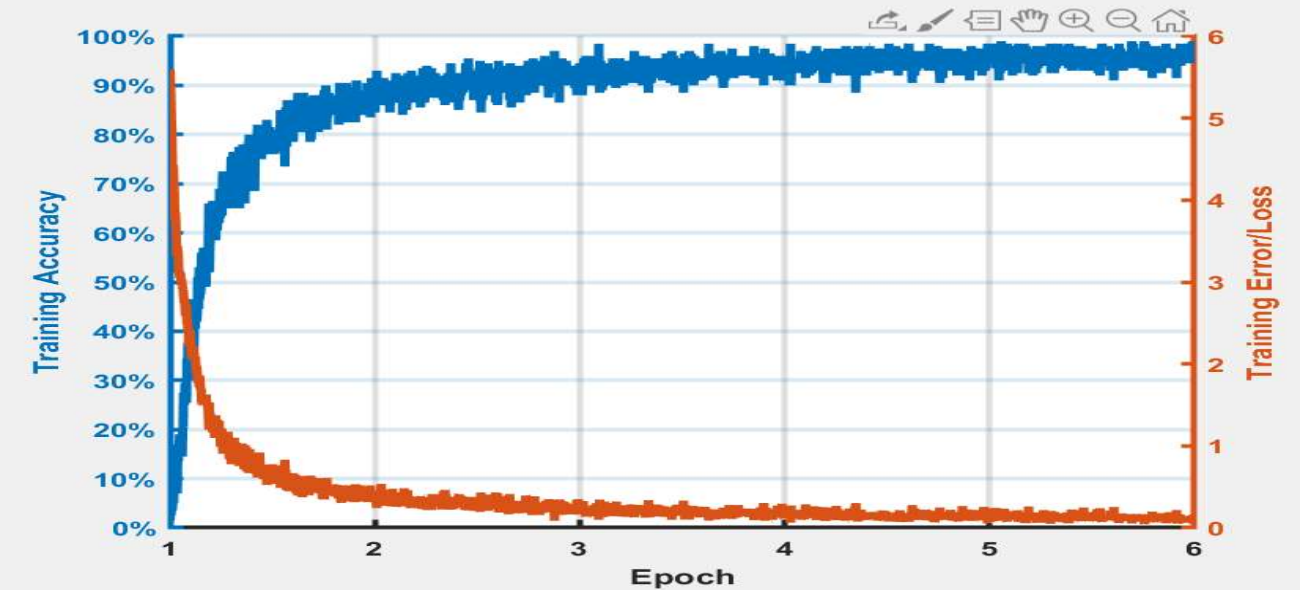


Figure 24 Accuracy and loss plot for training

Batch Size: 200

No of Epochs: 6

Input Image Size: 256*256*3 (Converted to 224*224*3 before training and testing)

Pre-Processing Steps: Augmentation, Thresholding for segmentation of image.

Learning Rate: 3E-4

Training Accuracy: 97.79%

Validation Accuracy: 96.93%

Architecture: GoogleNet

Data Split: 80%: 10%: 10%

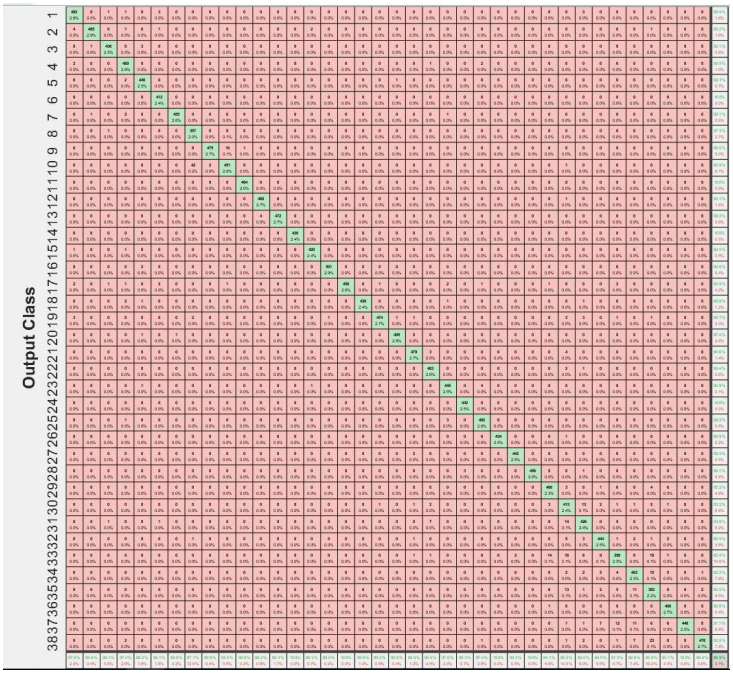


Figure 26 Confusion matrix for classification on training data

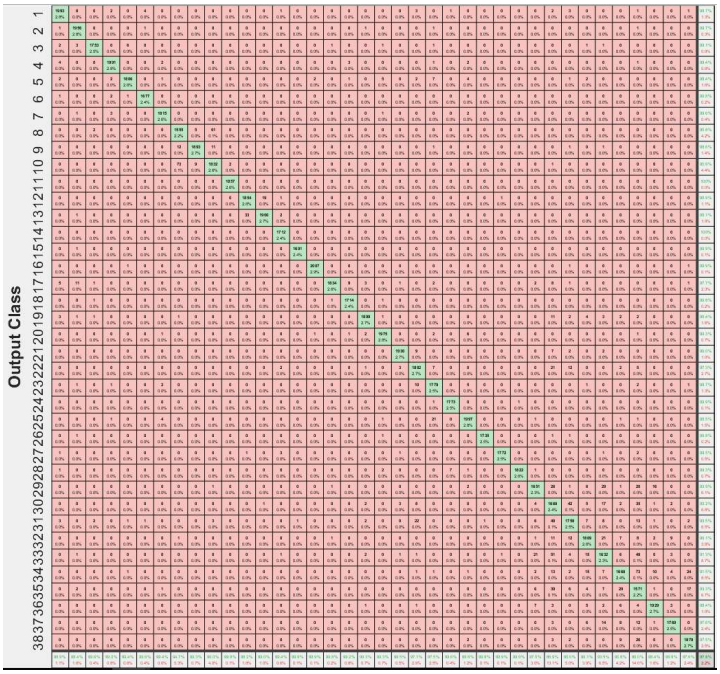


Figure 25 Confusion matrix for classification on validation data

We can observe that the accuracy is very much improved with the pre-processing steps followed before training the data for this architecture.

Which was an improvement from 85% of accuracy before processing and 96% accuracy after this step.

So we can say that the better features were being able to be detected by the network for training itself when we enhance the training images and increase intensity of the spots and supressing the background with low intensity pixels for segmentening them from the foreground leaf image.

The below are some of the visualizations of th trained netowork at this stage.

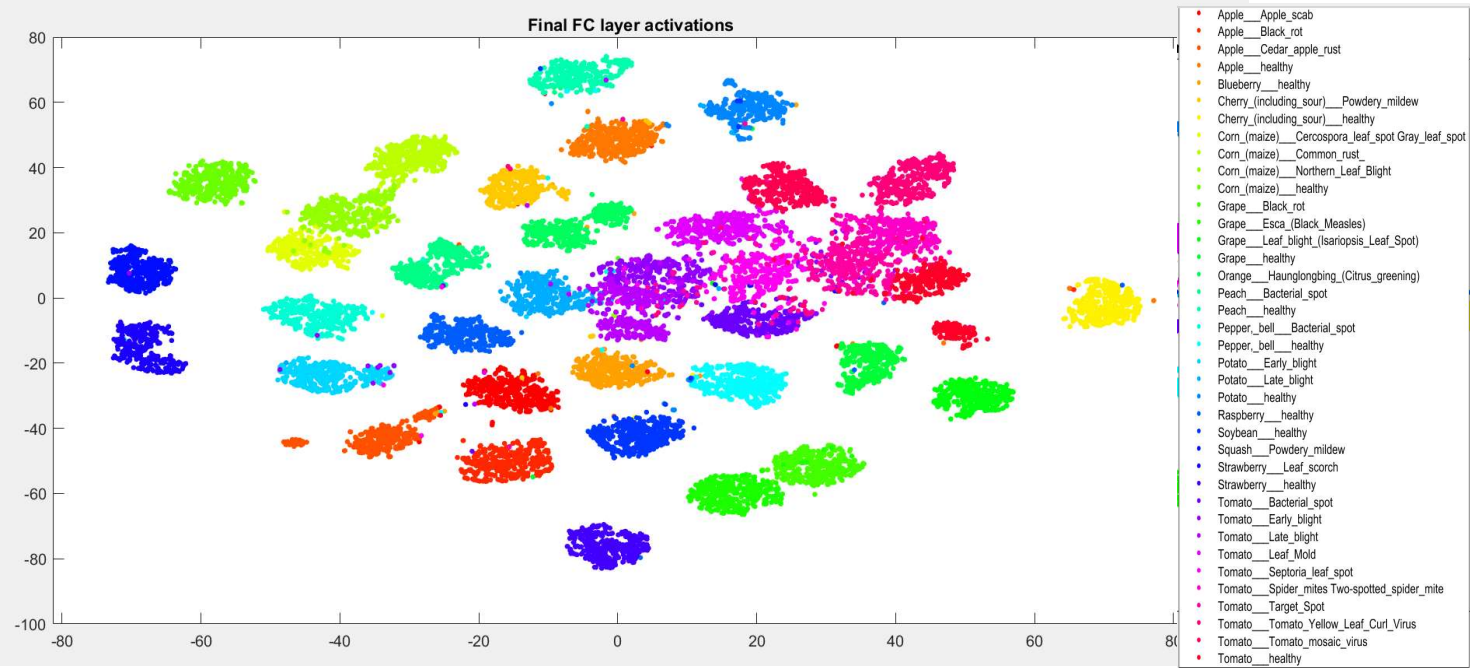


Figure 27 t-SNE visualization of the last FC layer of the network on the validation data

I have observed that with the AlexNet architecture the accuracy was less for some of the classes i.e. the performance of some particular were less and their dominance have decreased the overall accuracy of the model.

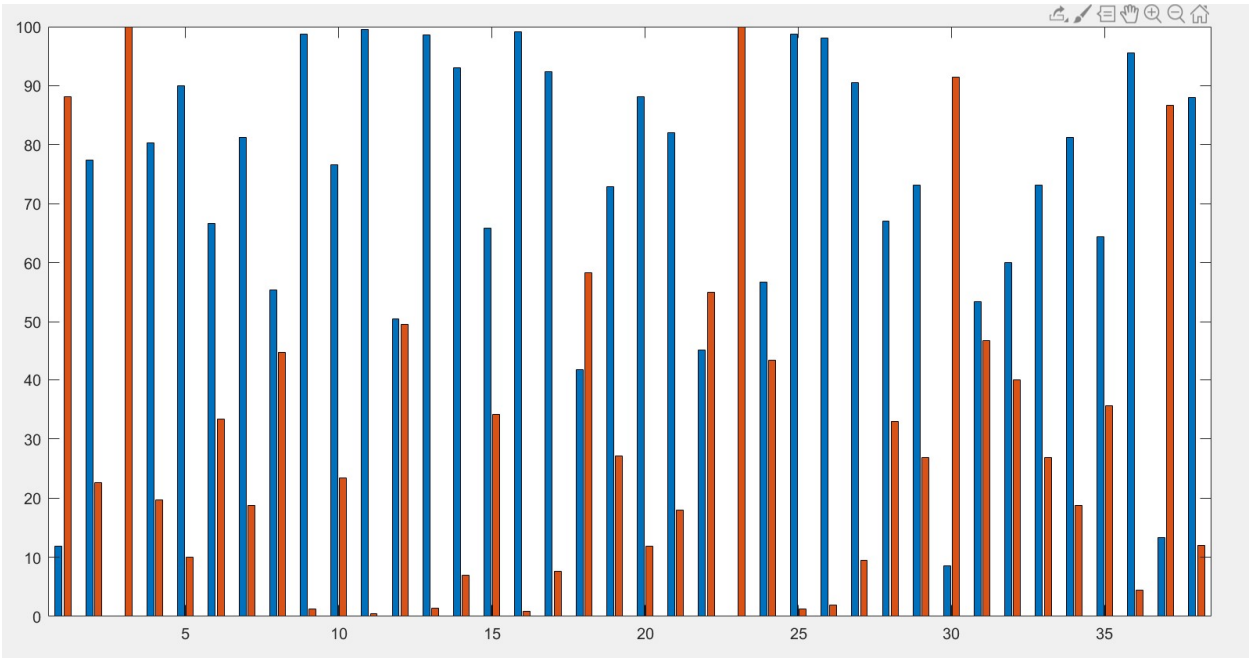


Figure 28 Class wise Accuracies along with failure rate.

Here we can see that some classes namely 1, 3, 18, 23, 30, 37 etc. have the failure percentage of classification of over 50%.

The reason for this behavior was due to the unequal number of images per class among all the 38 classes. i.e. some of the classes have 280 images and other have 1000+ sample images, in order to make them consistent and overcome this dominance due to irregularities between the samples per class I have augmented the entire data such that all the classes have equal number of samples.

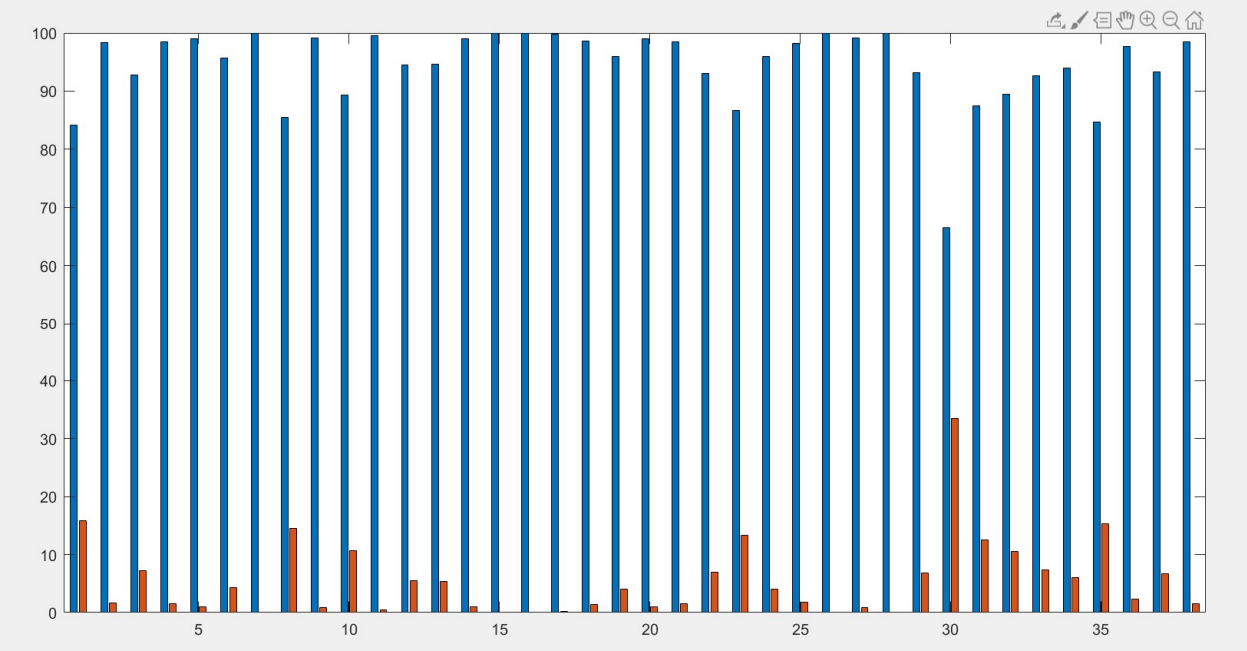


Figure 29 Class wise accuracies using GoogleNet Architecture

Then after These changes the visualization of the learned features were observed for this, I have generated the images which best resemble the features trained for determining that particular classes by iteration on the FC layers for the DeepDreamImage in MATLAB. The below are some of my interesting observations from these visualizations.

Figure 30 Corn leaf healthy

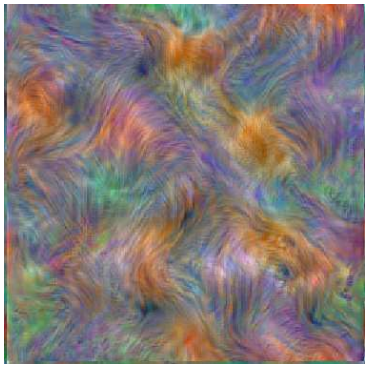


Figure 31 Trained features by network

By observing the above features, I conclude that the model had trained on the features of the leaf texture and the vertical lines on the leaves only. As the samples in the dataset for the corn healthy class are all zoomed in images and there is no possibility for the model to learn the leaf shape features for this class like the other classes, where the dataset contains the images of the entire images without zooming in, So I have concluded that this might be the issue for the failure of classification for this particular class.

Figure 32 Potato leaf healthy

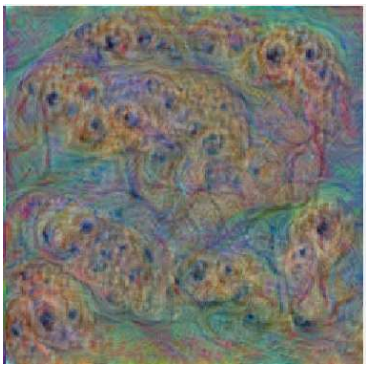


Figure 33 Trained features by the network

Figure 34 Grape Leaf Blight

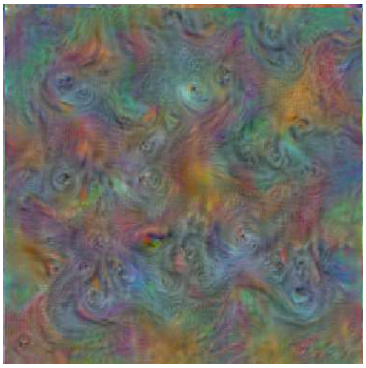


Figure 35 Trained features by network

Figure 36 Grape Esca

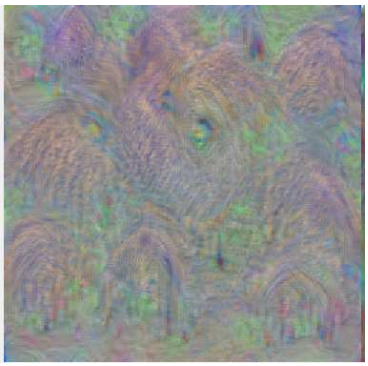


Figure 37 Trained features by network

From the above images we see that the network was trained features for some of the classes where the failure rate high than other classes. We see that the features of the spots and the leaf shape were trained unlike the corn. And there are many similarities for these features among most of the features so that might be one of the backdrops for the failures.

Overall, by observing all these experiments, the final best results obtained for this classification problem are shown below.

Batch Size: 200

No of Epochs: 20

Input Image Size: 256*256*3 (Converted to 224*224*3 before training and testing)

Pre-Processing Steps: Augmentation, Thresholding for segmentation of image.

Learning Rate: 3E-4

Training Accuracy: 99.93%

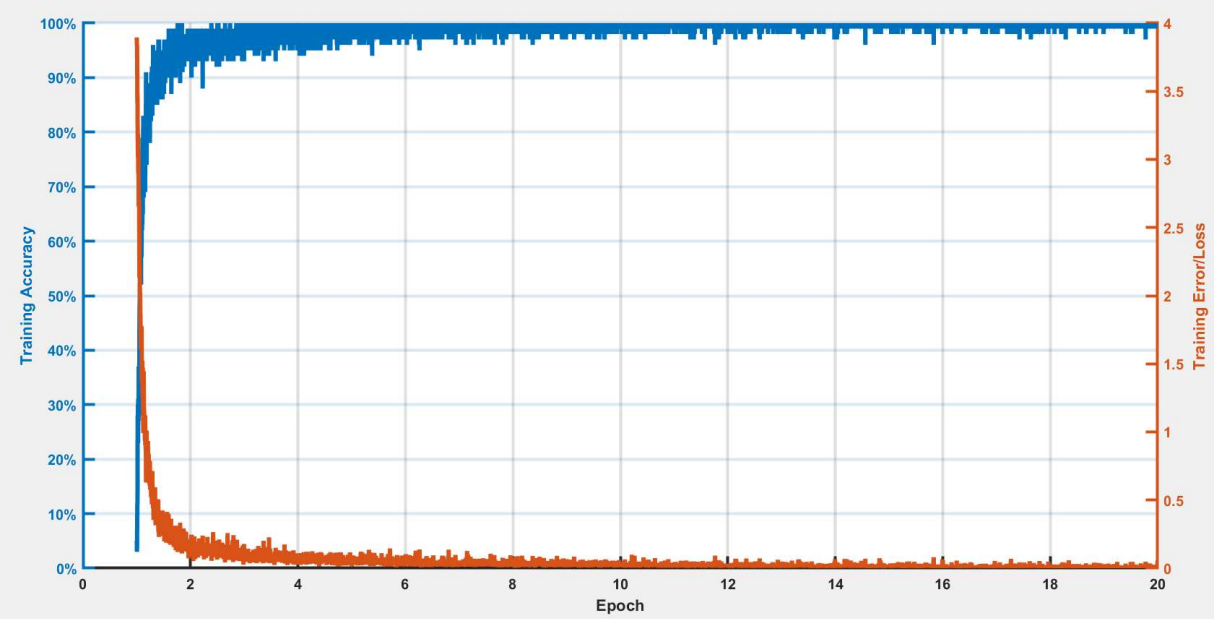
Validation Accuracy: 99.23%

Validation Accuracy: 99.12%

Network Architecture: ResNet50

Data Split: 70%: 15%: 15%

The best accuracy I have achieved with this dataset is 99.12% on the test data using the ResNet50 architecture. The above shown are the details and the confusion matrix and visualizations are shown below.



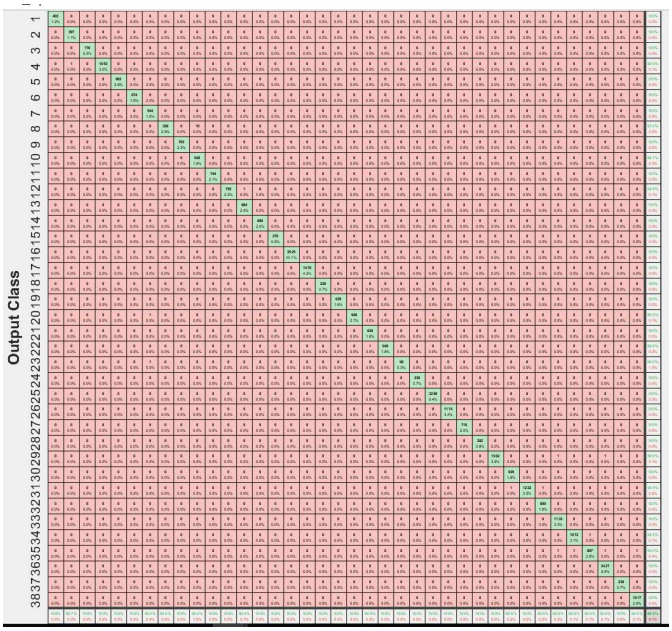


Figure 39 Confusion matrix on Train Data

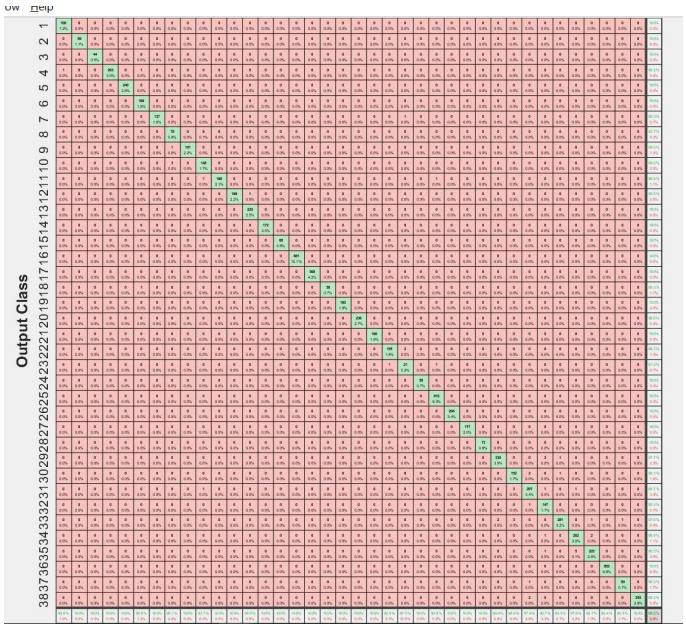


Figure 38 Confusion matrix on Validation data

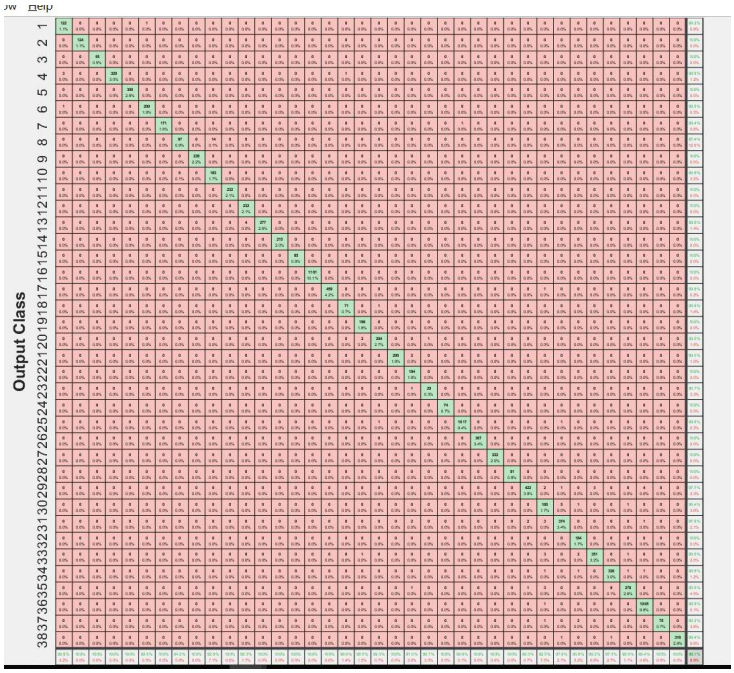


Figure 40 Confusion matrix on test data

The below are the class wise accuracies of this network, still we can see that there are some classes where the failure rates greater than 5%, Which we have also visualized in the above sections for the failed classes.

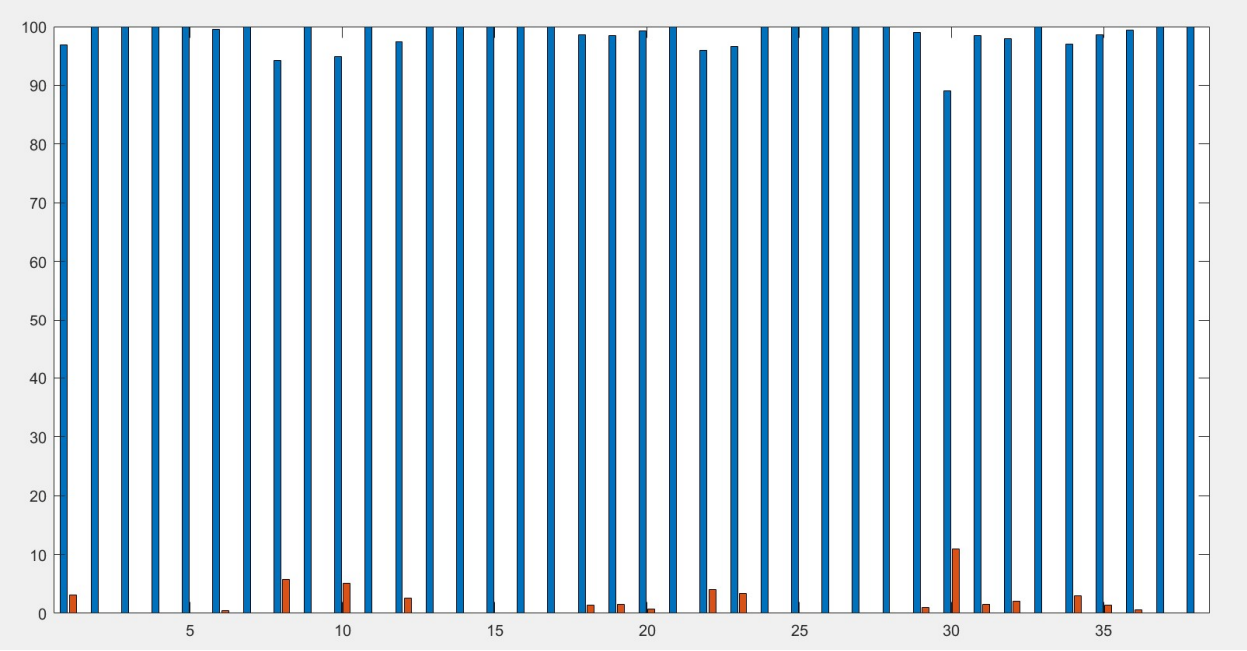


Figure 41 Class wise accuracies of the best performing network in this project

Final Results Tabulated:

Network	Train: Val: Test	Training Acc	Validation Accuracy	Test Accuracy
Basic CNN	60:20:20	58.96%	58.67%	58.73%
Basic CNN	70:10:20	55.23%	54.93%	52.86%
Basic CNN	50:20:20	54.22%	52.16%	51.59%
AlexNet	70:15:15	80.92%	80.05%	78.92%
AlexNet	70:10:20	84.23%	83.23%	81.78%
AlexNet	60:20:20	82.03%	81.50%	80.97%
AlexNet	50:20:30	81.68%	81.03%	80.62%
ResNet	60:20:20	99.92%	99.11%	99.09%
ResNet	70:15:15	99.94%	99.48%	99.21%
ResNet	70:10:20	99.93%	99.23%	99.12%

The below table is the mean accuracies and the standard deviations for the different architectures Train, Test & validation data for different splits.

Network	Training Acc	Validation Accuracy	Test Accuracy
Basic CNN (Mean)	56.1367 %	55.2533%	54.3933%
AlexNet (Mean)	82.2150%	81.4525%	80.5725%
ResNet (Mean)	99.9300%	99.2733%	99.1400%
Basic CNN (Std)	2.4967	3.2670	3.8090
AlexNet (Std)	1.4210	1.3301	1.2040
ResNet (Std)	0.0100	0.1888	0.0624

Conclusion & Future Scope:

From all of the above observations I can conclude that for this particular dataset the classification of 38 distinct classes was efficient with the deep learning techniques while I have also discussed the case where I got the very less accuracy with the HOG features extracted and classified.

We can also observe that there are only 5-6 classes which are being failed for around 5% of the classification for testing by which the accuracy is effected and by the observations of the visualizations we can conclude that they can also be improved by updating the dataset and extracting clearer images for those class samples also similar to all other classes.

We can also expand this classification problem further by adding up additional plant classes, The authors of the dataset are also keep adding new images to the dataset so we can expect this classification to become some complex as the number of classes increase while compared to 38 classes now.

We can also further improve the problem statement to work on adding the feature of detecting the diseased part of images instead of just detecting the disease, For which we also have to label the data for training and also as a future scope we can also add additional features to this machine learning approach to suggest the fertilizers to be given to plant for that disease along

with its detection as well, Where by feeding the leaf image the network will detect the disease and also gives the list of fertilizers etc.

As part of this project I have learned many different approaches for improving the accuracy of the network other the hyper parameters, And the importance of the data and its key role in the classification training is also understood practically alongside of this project. Different methods to extract the features from images, Pre-processing of data, Augmentation of data to make sample count equal across all classes and its importance is also practically observed in this project.

References

[1] Ramesh Maniyath, Shima & V, Vinod & M, Niveditha & R, Pooja & N, Prasad & N, Shashank & Ram, Hebbar. (2018). Plant Disease Detection Using Machine Learning. 41-45. 10.1109/ICDI3C.2018.00017.

[2] M. Sardogan, A. Tuncer and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," 2018 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 2018, pp. 382-385.

[3] B. Wang and D. Wang, "Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network," in IEEE Access, vol. 7, pp. 151754-151763, 2019.

[4] R. M. Prakash, G. P. Saraswathy, G. Ramalakshmi, K. H. Mangaleswari and T. Kaviya, "Detection of leaf diseases and classification using digital image processing," 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2017, pp. 1-4.

[5] N. G. Kurale and M. V. Vaidya, "Classification of Leaf Disease Using Texture Feature and Neural Network Classifier," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, 2018, pp. 1-6.

[6] S. Widiyanto, R. Fitrianto and D. T. Wardani, "Implementation of Convolutional Neural Network Method for Classification of Diseases in Tomato Leaves," 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 2019, pp. 1-5.

[7] H. Durmuş, E. O. Güneş and M. Kırcı, "Disease detection on the leaves of the tomato plants by using deep learning," 2017 6th International Conference on Agro-Geoinformatics, Fairfax, VA, 2017, pp. 1-5.

[8] Loui, Alexander, Zhang, Keke, Wu, Qiufeng, Liu, Anwang,Meng, Xiangyan "Recent Machine Learning Progress in Image Analysis and Understanding" Hindawi, Advances in Multimedia, 10.1155/2018/6710865.

[9] https://medium.com/@evergreenllc2020/fundamentals-of-image-thresholding-and-masking-6a89997ccca6

[10] https://www.mathworks.com/discovery/image-thresholding.html

[11] Mohanty Sharada P., Hughes David P., Salathé Marcel "Using Deep Learning for Image-Based Plant Disease Detection" Frontiers in Plant Science V7 2016 10.3389/fpls.2016.01419 ISSN 1664-462X.