Plant Leaf Disease Detection and Classification

Sakshi Saini
2017092

Priya Rajpurohit
2015073

I. Problem

The problem is to identify disease by analyzing the leaf from a plant. The dataset consists of images of healthy and diseased leaves. The situation we are trying to handle 1) prediction of whether a plant leave is diseased or healthy 2) If found diseased accurately identify between many diseases.

II. Motivation

Agriculture has a vital role in the life of an average human being, or it is something that we all depend on. India has always been an agrarian powerhouse. Agriculture is something that the Indian economy profoundly depends on. The foremost obstacle faced by an agronomist is the disorders in plants.
Diseases in plants are prevalent in agro-industry. Ignoring the health of crops and not giving them proper attention and care may lead to low standard products, and the efficiency of the farmer is affected.
When modern technologies like deep learning and computer vision are being put to use, in the detection of any crop disease in an advanced phase, they can be treated and prevented from getting wasted. Hence, increasing the net productivity. This project is just an attempt to see how various approaches in Deep learning can be used and evaluate their effectiveness in detection. However, the diseases in focus are mainly found in plant leaves.

III. Literature Review

When a plant leaf is identified as infectious/infected, the primary issue next is to diagnose it further. One needs to find out the disease it got infected by and steps to tackle it effectively and prevent further spread to others. Otherwise, a considerable loss occurs for the farmer- social, economic, or ecological.
Earlier, methods of controlling this situation were manual and tricky. However, nowadays, with affordable mobile phones having inbuilt cameras are readily available almost everywhere. This advancement has turned out to be a significant blessing in agriculture since it has facilitated disease diagnosis tools to be used by anyone and at no expense. In the preceding few decades, there has been much progress made in deep learning and computer vision. It is now possible to classify an image based on trained models in a few seconds only.
Classifiers are needed to be trained on prominent features of plant leaves such as - color, shape, and texture to attain better disease classification. CNN's have been used for classification earlier. However, for achieving higher accuracy, various feature extraction, preprocessing techniques, image datasets, and cross-validation functions have already been applied. Nevertheless, deep learning architectures which have lately been introduced, are said to show better performance than a generic CNN architecture.
This paper will be presenting a comparison of various deep learning architectures that have been used for diseased leaves classification in the past. The architectures that will be examined are- AlexNet, GoogleNet, VGGNet, ResNet, Inception, and Xception. One must be able to detect if a plant leaf is diseased or not to classify the kind of disease that leaf has. These are just a few issues; we will try to compare the mentioned architectures.
To train various models to classify diseases for proper diagnosis accurately, a large verified dataset of images of diseased as well as healthy plants are needed. Until recent times, such datasets did not exist, and the few small datasets available were not freely available. PlantVillage is a project that curated a dataset of healthy and diseased plant leaves, which is open and free to use.

A. GoogleNet

GoogleNet was the winner of ILSVRC 2014; it was able to achieve a top-5 error rate of 6.67%. It was acknowledged to be the most suitable architecture proposed in all the previous CNN architecture submissions. This winning neural network implementation was based on an approach inspired by LeNet. They proposed the concept of the famous Inception module, the basic idea of the module was to cover a large section of an image while simultaneously capturing small details like edges, texture, shapes with excellent resolution. It consists of 22 layers of convolutional layers that were successful in reducing the parameters from 60 million to 4 million, which was a great achievement in itself.

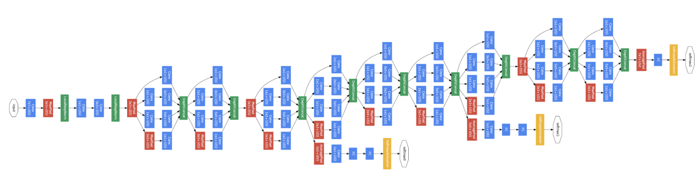

Fig. 1. GoogleNet Architecture

B. AlexNet

The architecture consists of 8 layers: 5 convolutional layers and two fully-connected hidden layers and one fully connected

output layer. AlexNet famously conquered the 2012 ILSVRC-2012 competition by reducing error rates to 15.3%.
The use of ReLU added a non-linearity factor to this architecture, which The use of ReLU added a non-linearity factor to this architecture, which resulted in the acceleration of the computation speed six times. Overfitting was handled by the use of dropout(0.5) instead of ordinarily used regularization methods. Overlap pooling was adopted to reduce the size of the network significantly.
The architecture decreases the number of features from 227*227*3 to 1000. The network has 62.3 million parameters and needs 1.1 billion computation units in a forward pass. The convolutional layers account for 6% of all the parameters, consumes 95% of the computation.

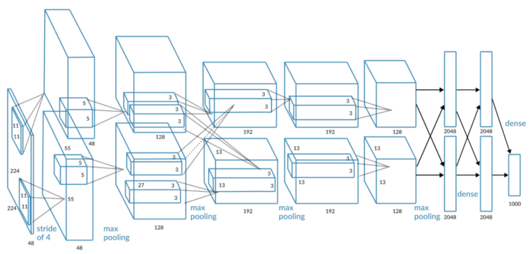

Fig. 2. AlexNet Architecture

C. ResNet 50

He et al. first presented ResNet in their 2015 paper, Deep Residual Learning for Image Recognition. ResNet architecture has become a groundbreaking work, illustrating that extremely deep networks can be trained using standard SGD (and a reasonable initialization function) through the use of residual modules.
With the help of residual blocks, it increases the number of hidden layers as much as one aspires without worrying about the vanishing / exploding gradients problem. Residual blocks facilitate the network to preserve what it had learned previously (if there is nothing to learn). It is done by having an identity mapping weight function, which preserves what the neural network has learned by not employing diminishing transformations. If the layer can learn something, it will add to what the network has already learned.
ResNet was initially introduced with 152 layers, but ResNet 50 is more popularly utilized because it takes too much time to train Resnet 152 as opposed to Resnet 50.

D. Inception v3

Szegedy et al. first introduced the "Inception" micro-architecture in their 2014 paper, Going Deeper with Convolutions..
With multi-model multi-crop, Inception-v3 with 144 crops and four models ensembled, the top-5 error rate of 3.58% is achieved and finally secured 1st Runner Up (image classification) in ILSVRC 2015.
The significant upgrades in this version were- use of rmsprop

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 3. ResNet 50 Architecture

optimizer, factorized 7*7 convolutions, batch normalization in auxiliary classifiers, and label smoothing.

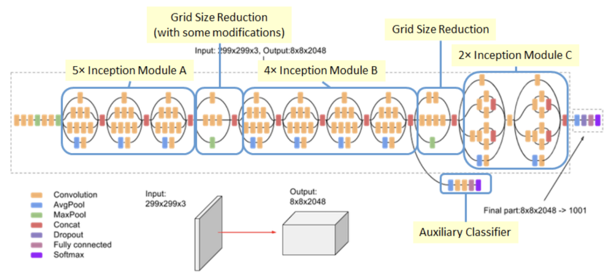

Fig. 4. Inception v3 Architecture

E. Xception

Xception somewhat defeats Inception v3 on the ImageNet dataset and hugely beats it on a larger image classification dataset with 17,000 classes. Most importantly, it has the same number of model parameters as Inception, implying a more elevated computational efficiency. Xception is much newer (2017). Xception was submitted by Francois Chollet.
Xception is an extended version of the Inception architecture which replaces the standard Inception modules with depthwise separable convolutions followed by pointwise convolution. It is based on the hypothesis that: spatial correlations and cross-channel correlations can be appropriately decoupled.

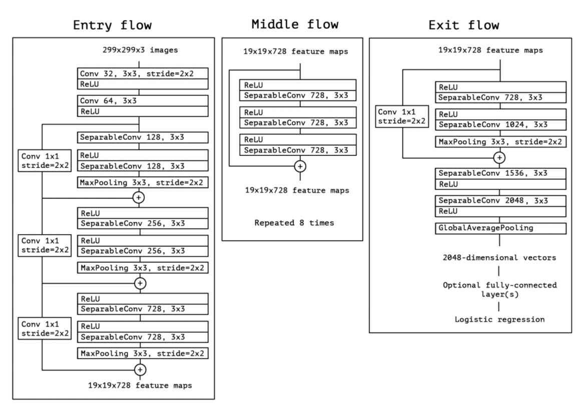

Fig. 5. Xception Architecture

F. VGG16

Simonyan and Zisserman presented the VGG network architecture in their 2014 paper, Very Deep Convolutional Networks for Large Scale Image Recognition.
This network is distinguished by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. The reduction of volume size is handled efficiently by max pooling. It has two fully-connected layers, each with 4,096 nodes, which are then followed by a softmax classifier.

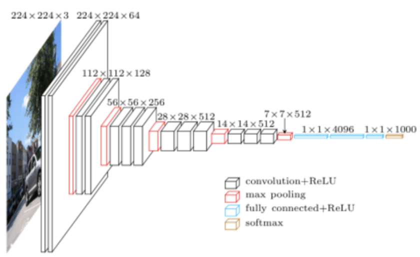

Fig. 6. VGG16 Architecture

IV. Dataset Details

A. About Dataset

The PlantVillage dataset consists of 54303 healthy and diseased leaves images divided into 38 classes by species and disease. The dataset is available in TensorFlow datasets.
Original paper: https://arxiv.org/abs/1511.08060
Dataset: https://www.tensorflow.org/datasets/catalog/plant_village

B. Data Visualization

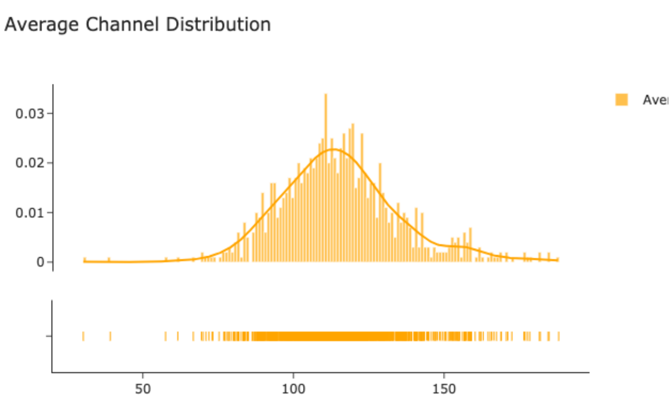

Fig. 7. Average Color Distribution

The plot of the average of all channel distributions is almost similar to a Gaussian distribution. The mean of the plot is around 110, and the standard deviation is approximately 30. The maximum achievable activation is 255. Images are minimally active most of the time.

1) Red Color Distribution: Significant fluctuations can be seen in the activation of red color in the dataset. It can be observed that it shows a plot similar to Gaussian distribution having a mean around 100 and a standard deviation of 40 with distortion at the left.

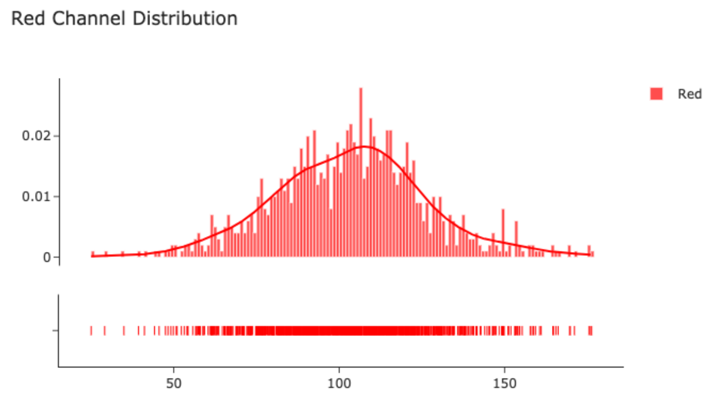

Fig. 8. Red Color Distribution

2) Green Color Distribution: As can be seen from the plot, green color is more uniformly distributed in contrast to red color distribution. It can be observed that the mean is around 120, and the standard deviation is approx. 30. It can be inferred that green color is more activated than red color in the dataset, which corresponds to the fact that most parts of any leaf are green.

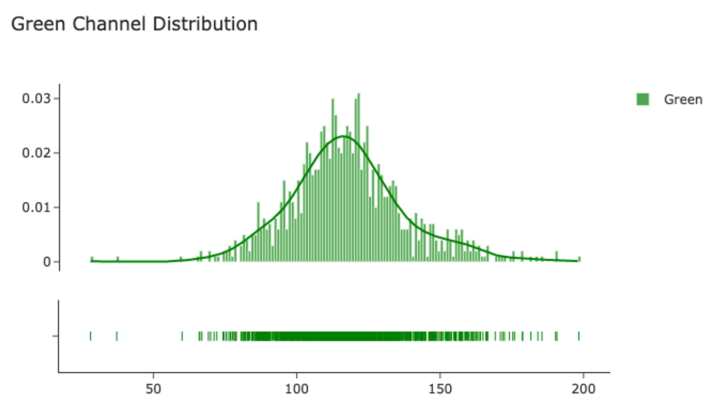

Fig. 9. Green Color Distribution

3) Blue Color Distribution: The plot of the blue color is similar to that of green color but slightly moved towards the left. A small distortion can also be observed on the right side of the curve.

Some published papers concluded that the observed skewness in plots of the distribution of three colors in the models suffers from some drawbacks. A similarity to that

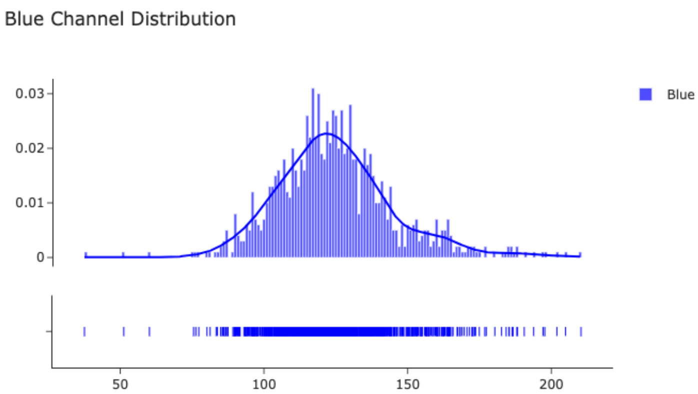

Fig. 10. Blue Color Distribution

can be seen in the red color distribution. This skewness can be beneficial for observing changes in the homogeneity of leaves(images).
https://www.researchgate.net/publication/339510794_Skewed_distribution_of_leaf_color_RGB_model_and_application_of_skewed_parameters_in_leaf_color_description_model

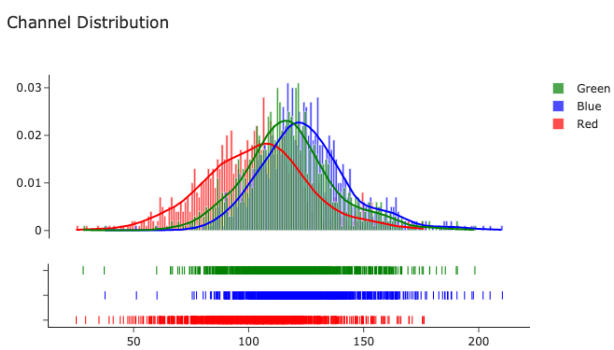

Fig. 11. RGB Color Distribution

V. Proposed Architecture

The chosen dataset has images of shape 256*256*3; therefore, we were required to find models with a similar input shape. We then proceeded to explore models that have shown relatively good outcomes on datasets with comparable characteristics.
The search for the optimal dataset stopped with Imagenet Large Scale Visual Recognition Challenge(ILSVRC). ImageNet is an extensive research project for research on the utility of visual object recognition. ILSVRC is organized by ImageNet every year. This challenge is a platform for programs to compete for precisely classifying and recognizing objects and scenes. The standard dataset for this challenge has 1000 classes. Thus we assumed that these models would train well on PlantVillage, which has mere 38 classes.
On doing more research, we found the six deep learning architectures that could train successfully on PlantVillage. The reason being that each one of them was either a winner or runner-up of ILSVRC in various years. We also decided to use pre-trained models of ResNet50, Inception v3, VGG 16, and Xception; for there were computational limitations present in Google Colab.
We attempted to experiment with both approaches- 1) using architectures (ResNet50, Inception v3, VGG 16, and Xception) as feature extractors and then classify using Classifiers: SVM and Random Forest. And 2) using the technique of transfer learning and adding few more trainable layers to them for multi-class classification. Alongside using them for training as well as classification.

A. Pre-processing

We initially trained the models on colored images and grayscale versions of them, but when the images were processed using the methodology of image segmentation to separate leaves from the background. The process was to make a boundary around the biggest object present there in the image, followed by masking and retaining the leaf image. The general idea was to remove noise from images, which otherwise would have affected training models. On doing this, a slight improvement was observed in the accuracy of the classification using AlexNet and GoogleNet. Therefore, we used the segmented images only for training models, moving forward.
The need to use PCA on the image for preprocessing was not felt, given that we were already using segmented images. Keras applications library provides numerous functions, including the ones for preprocessing images specific for each model. (eg. keras.applications.xception) This was applied to images to make sure they were already normalized and would give better results with stated architectures.

B. Multi-Class Classification and Models

The main aim was to classify images based on whether leaves were healthy or diseased if found diseased, identify the type. In the dataset, 38 classes have been defined, among which 12 are healthy, and rest are diseased.
We wanted to examine the following two scenarios:

1) When we only have a small dataset with insufficient information, these dense models require a large dataset to obtain a better performance.
2) When we have a dataset, large enough- then how can we manage to train models without getting the model too complicated.

Let us consider scenario 1; in such a case, we have tried to perform the transfer learning method and utilize them to classify them, using classifiers - SVM and random forest. The models mentioned above are well recognized and are trained to extract features like curves, edges, color, shape, and

texture. We can use any classifier that fits best according to the PlantVillage dataset.
In the second scenario, we have used untrained models like AlexNet, GoogleNet, and train them on PlantVillage dataset. Otherwise, we have used transfer learning using the ImageNet pre-trained weights to extract features from images(for Xception, Inception v3, ResNet 50, and VGG16). For doing so, we have frozen the layers of that model except for the last few ones.

C. Binary Classification

Focusing on the other problem specified earlier, we now had to classify an image of a leaf as diseased or healthy(binary). We tried to solve this by using the label names of the 38 classes. If a name had substring 'healthy' in it, we classified it as healthy and otherwise diseased.

VI. Visualizations

1) Color images of Leaves in the dataset

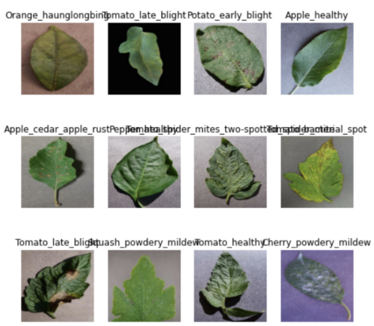

Fig. 12. Color images

2) Grayscale images of Leaves in the dataset

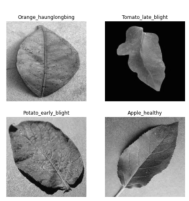

Fig. 13. Grayscale images

3) Segmented images of Leaves in dataset

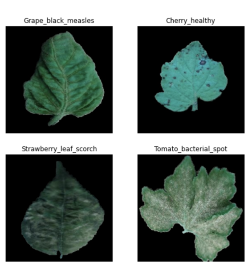

Fig. 14. Segmented images

4) GoogleNet model trained on color images

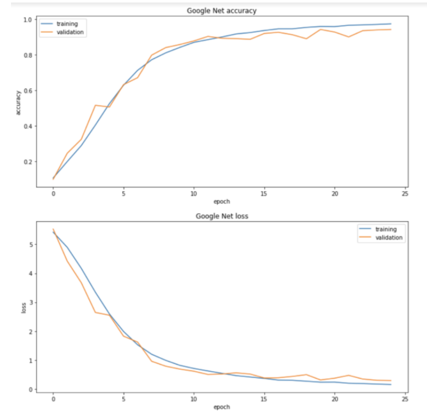

Fig. 15. GoogleNet on color images

5) GoogleNet model trained on grayscale images

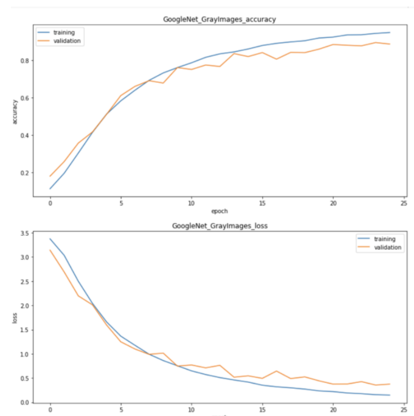

Fig. 16. GoogleNet on grayscale images

6) GoogleNet model trained on segmented images

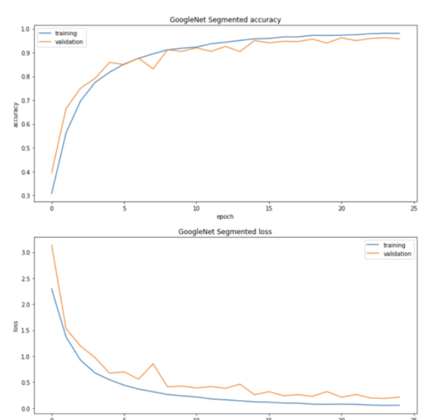

Fig. 17. GoogleNet on segmented images

7) AlexNet model trained on color images

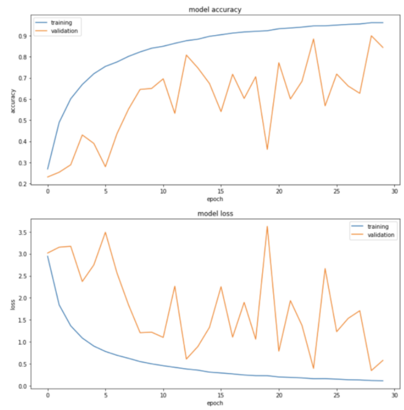

Fig. 18. AlexNet on color images

8) AlexNet model trained on grayscale images

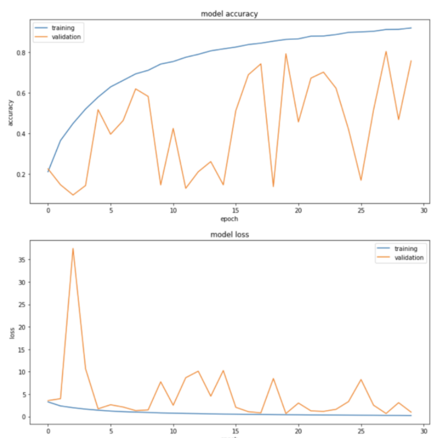

Fig. 19. AlexNet on grayscale images

9) AlexNet model trained on segmented images

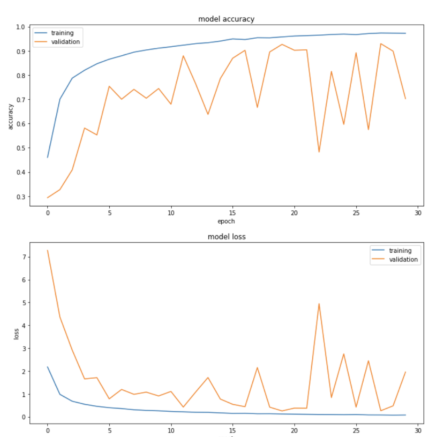

Fig. 20. AlexNet on segmented images

10) Inceptionv3 model trained on segmented images

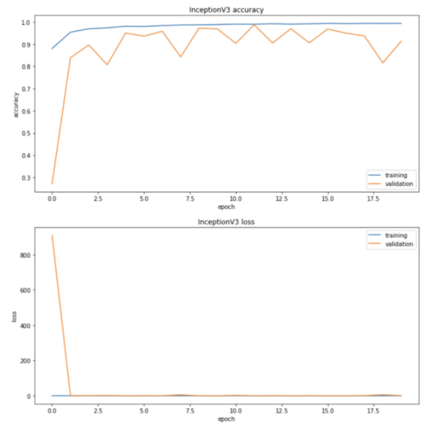

Fig. 21. Inceptionv3 on segmented images

11) Xception model trained on segmented images

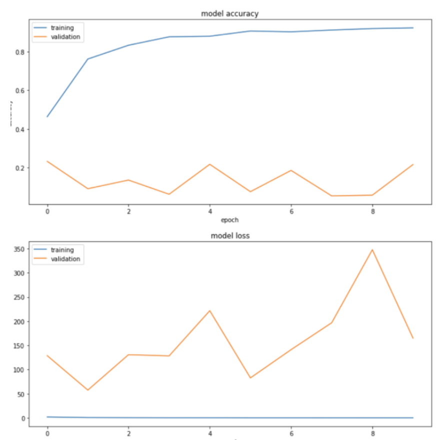

Fig. 22. Xception on segmented images

VII. Results

S.No.	Model	Accuracy on Classification
1	AlexNet - trained on PlantVillage dataset	On segmented images- 0.799 On color images-0.749 On grayscale images-0.543
2	GoogleNet - trained on PlantVillage dataset	On segmented images-0.958 On color images-0.942 On grayscale images-0.885
3	ResNet 50 - pre-trained on ImageNet dataset (feature extractor)	On segmented images Classified using SVM-0.846 Classified using RF-0.826
4	Inception v3 - pre-trained on ImageNet dataset (feature extractor)	On segmented images Classified using RF-0.821
5	Xception - pre-trained on ImageNet dataset (feature extractor)	On segmented images Classified using SVM-0.978 Classified using RF-0.777
6	VGG16 - pre-trained on ImageNet dataset (feature extractor)	On segmented images Classified using SVM-0.981 Classified using RF-0.903

VIII. Analysis of Results

A. Feature Extractors

Models, having VGG 16 and Xception as feature extractors and SVM as classifiers has outperformed all the other trials in this experiment. The original shape of the images is 256*256*3. Post feature extraction, the dimensionality of the image reduces to 4096 for VGG16 and 7*7*2048 for Xception. The models performed better than others because of the following reasons. The models such as AlexNet or ResNet train on millions of parameters, so training such dense models is a pretty hectic task. If we use VGG16 or Xception as feature extractors, we can use simple models like SVM and Random forest, whichever works better with the dataset.

B. Classifiers

Transfer Learning is preferred when one does not want to train millions of parameters which try to learn edges, curves, color, the contrast of images. This is possible because these parameters remain the same for many image datasets.
GoogleNet turned out to perform better than AlexNet in terms of computational speed and accuracy; since it had to train fewer parameters, resulting in a smoother curve.
Xception gave a higher accuracy as a classifier, but it failed on test data as compared to VGG16 and GoogleNet. The higher accuracy is because it has the highest number of layers, and extracts much better features than ResNet50 and Inception v3. The principal problem with convolutional models is that they try to overfit datasets, hance, fail on testing, which can be observed from the curves plotted(see Visualizations). Thus, GoogleNet and AlexNet perform pretty well on training data but mispredict for unseen data.
The same overfitting is a significant reason for the failure of ResNet50 and Inceptionv3 on this particular dataset. For these models, this happened because both these models have a high number of layers as compared to others.

C. Binary Classification

The binary classification(healthy/diseased) was tested on two models, VGGNet 16 and Xception. The accuracies reached were- 0.993 and 0.997, respectively. The reason for such high accuracies was the high accuracies of the models when the classifier used along with the models was SVM.

IX. Inferences and Conclusion from results

It can certainly be concluded that combinations of VGG16 and SVM, Xception, and SVM are capable of classifying plant leaves diseases with comparably high accuracy than all the other models considered in this report. Apart from these two models, GoogleNet was able to produce a high accuracy, considering that it was not pre-trained.

X. Contributions

Contribution by Sakshi(2017092)

- GoogleNet
- Image Segmentation
- Data Visualization
- VGG16
- Inceptionv3

Contribution by Priya(2015073)

- AlexNet
- ResNet50
- Xception
- Binary Classification
- Inceptionv3

Contribution by both

- Presentation

References

[1] https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/
[2] AlexNet Code: https://engmrk.com/alexnet-implementation-using-keras/?utm_campaign=News&utm_medium=Community&utm_source=DataCamp.com
[3] GoogleNet Code: https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/
[4] Going Deeper with Convolutions - https://arxiv.org/abs/1409.4842
[5] ImageNet Classification with Deep Convolutional Neural Networks - http://www.cs.toronto.edu/~hinton/absps/imagenet.pdf
[6] Deep Residual Learning for Image Recognition - https://arxiv.org/abs/1512.03385
[7] Rethinking the Inception Architecture for Computer Vision - https://arxiv.org/abs/1512.00567
[8] Very Deep Convolutional Networks For Large-Scale Image Recognition - https://arxiv.org/pdf/1409.1556.pdf
[9] Xception: Deep Learning with Depthwise Separable Convolutions - https://arxiv.org/abs/1610.02357

Link to Drive folder having models: https://drive.google.com/drive/folders/1XoGfBflhmlIBX4G4MMDk-k36QP7xMHkC?usp=sharing