# Classification Problem

Syed Uzair 1802020, Zeemal Urooj 1802019, Sajid Hamid 1802013

Namal Institute
Computer Science

July 19, 2021

## Abstract

We have given a problem to identify between healthy pepper leaf and unhealthy pepper leaf. We implemented two neural network architectures to classify these two classes. First classifier is AlexNet Classifier in which there are 5 convolutional layers and second one is our own classifier in which there are two convolutional layers. We used a data set of pepper to train our models and then used raw data of images to test our models. In this implementation, we analyse how we can make a better neural network and what are key factors on which a neural network depends on.

# Contents

## 0.1 Classifiers

We used two neural network architectures.

- AlexNet Architecture
- Our own Architecture

### 0.1.1 Our Classifier

We have to make a classifier to classify the healthy and bacterial pepper leafs. This classifier was our own classifier. It was a binary classification problem because we have two classes, one of health and second of unhealthy pepper plants. We divided our data into two parts. First was our training data and second was testing data. We trained our model on training data and test the model on testing data.

#### Layers

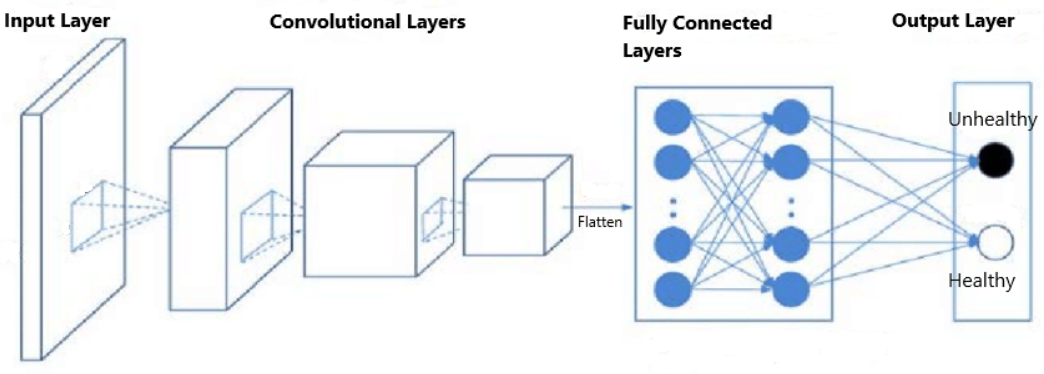


Figure 1: Our Own classifier model

## Activation Function

We used two activation functions in our own classifier. One is relu and other is softmax function.

A Gentle Introduction to the **Rectified Linear Unit (ReLU)**, in a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input.

The **softmax function** is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

## Accuracy

During the training of model, accuracy of our classifier is **99 percent**. Loss value during the training is **0.0039**. During testing of the model, accuracy of our classifier is almost 50 percent.

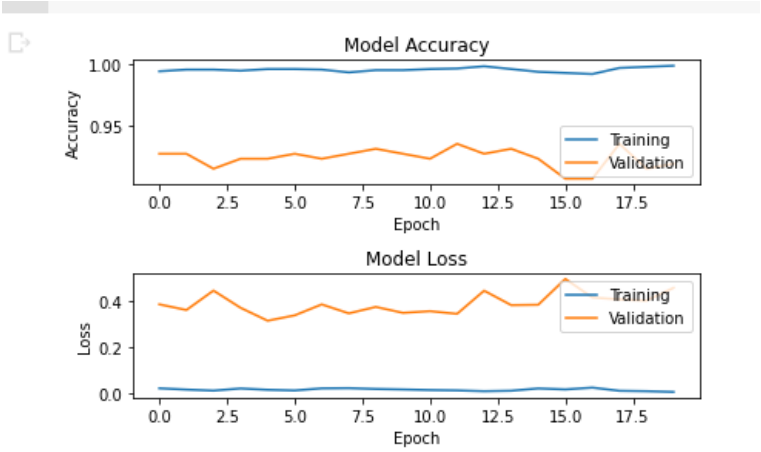


Figure 2: Accuracy and Validation

### Testing on Unseen Data

We used some pictures of health pepper leafs and some of unhealthy pepper leafs, which we obtained from google as raw data. Our classifier successfully classified the pictures of both healthy and unhealthy leafs. Same testing data was used in AlexNet architecture also.

## 0.1.2 AlexNet Classifier

AlexNet is a **convolutional neural network** which was primarily designed by Alex Krizhevsky And published by Ilya Sutskever and Krizhevsky's doctoral advisor Geoffrey Hinton.
This neural network consists of hidden layers such as convolutional layers, pooling layers, fully connected layers and normalizing layers. Convolutional is basically a process to apply a filter on the image to modify it.

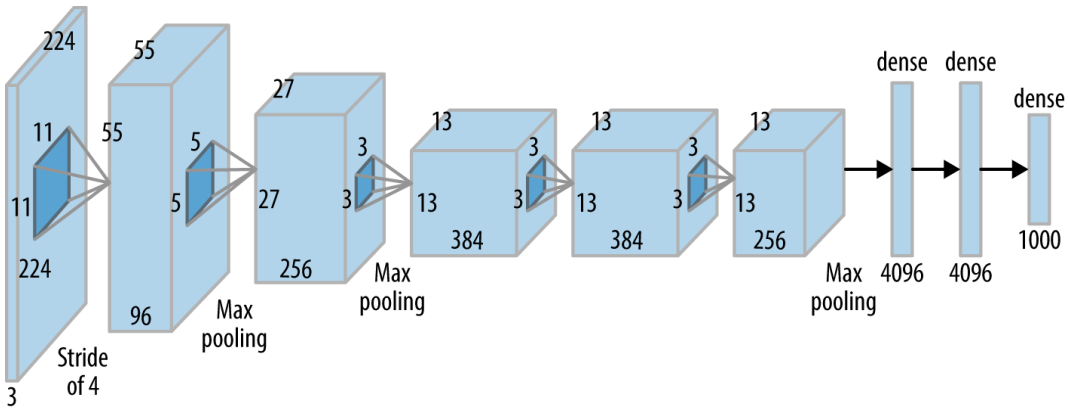


Figure 3: AlexNet Architecture

Here is the architecture of AlexNet neural network . It consists of **5 convolutional** layers, three max-pooling layers and two normalizing layers, two fully connected layers and one softmax layer. Each

convolutional layer consists of a filter and an activation function which is called relu.

Our task was to use AlexNet architecture to classify between two classes. First class is of **healthy Pepper leaf** and second one is of **unhealthy Pepper leaf**. We used AlexNet architecture to train our model for these two classes. We picked up the data set of both classes from kaggle.com. They have a vast amount of data sets which we can use in various classifications. Data set which we downloaded from kaggle.com consist of **1478** images of healthy pepper leaf and almost **1000** images of unhealthy pepper leaf.

### Working

When we train our model on the provided data set of pepper leaf, we fixed the size of epochs to **50** during the fit function. But surprisingly we obtained an accuracy of **100 percent in just 6th epoch**. Total number of features which our model extracted from data set are 78. We also plot a graph analyze the behavior of accuracy which is shown below.

### Accuracy on Testing Data

We saved our model and then used that saved model to test our data-set. First when we run the model on test data set it gives an accuracy of **0 percent** but when we changed the data set a little bit, accuracy increases. In our second turn we obtained and accuracy of **45 percent** and value of loss was 2.6. Third time with some addition of more images in the test data, we obtained an accuracy of **55 percent** and value of loss was 1.18. We analyse that with a better testing data set,

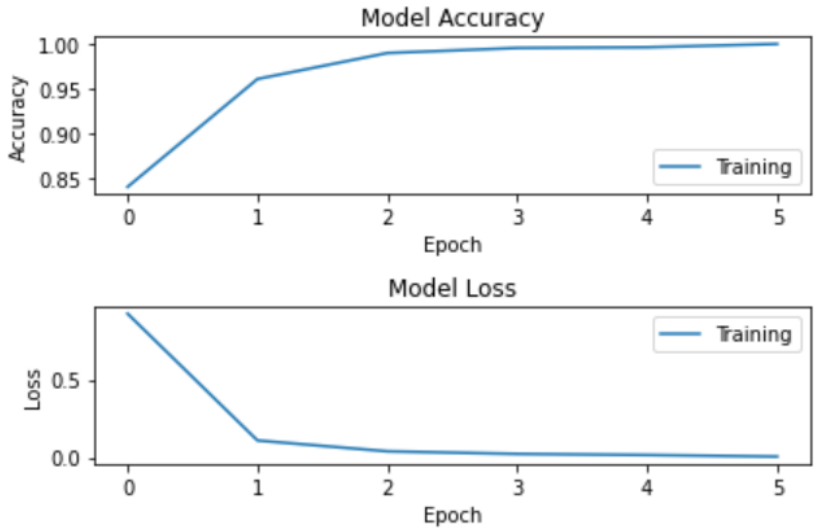


Figure 4: Accuracy and Loss

the accuracy can increase up-to 80 percent. We have to test our model on raw data of google images so unfortunately there was not enough suitable images we found. From the graph we can clearly analyze that with every epoch our measure of accuracy increases and our measure of loss function decreases and when we reach to the seventh epoch we obtained a flatten surface showing hundred percent accuracy and the value of loss function to Zero.

## 0.2 Comparison between Classifiers

We implemented two architectures of neural networks first one is our on architecture of neural network and the second one is AlexNet neural network architecture. We trained two models using these two architectures. During the training, we passed same data sets to both of architectures but our own neural network gives an accuracy of **70 percent** on training data set on the other hand AlexNet architecture

gives an accuracy of 100 percent during the training of model and another plus point for AlexNet architecture is, it gives an accuracy of 100 percent in juts 6 epochs.

I think the reason is, difference between numbers of hidden layers between both architectures. Our architecture uses two convolutional layers and in AlexNet architecture, there are total 5 convolutional layers. While training models of both architectures, AlexNet architecture achieve an accuracy of 100 faster than our own architecture of neural network.

## 0.3 Result

From our all calculations, we find out that AlexNet architecture performs well in comparison with our own implemented architecture. And it can be easily seen during the training phase of both models and by the results which we mentioned above.

## 0.4 Conclusion

We concluded that there are two main factors on which a neural network depends on. First one is number of hidden layers and second one is the data-set which we use while training as well as while testing. If we have sufficient amount of hidden layers in our architecture we can easily develop a good neural network. And if we have a filtered and sufficient amount of data set, then we can train our model well.

## 0.5 References


- AlexNet Implementation, Retrieved July 18, 2021, from https://towards alexnet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98
- AlexNet, Retrieved July 18, 2021, from https://medium.com/analytics-vidhya/alexnet-tensorflow-2-1-0-d398b7c76cf
- Google Testing Images(raw data), Retrieved July 18, 2021, from www.google.com
- Pepper Leaf Data set, Retrieved July 18, 2021, from www.kaggle.com