**Stage 3**

**Project Report**

**Proposed Title**: Classification of New Mexico Chile pepper plant disease using Multilayer Neural Network model.

**Introduction**

The severity of diseases caused by pathogens varies from mild symptoms to decline of the infected plants, depending on the aggressiveness of the pathogen, host resistance, environmental conditions, duration of infection and other factors. Plant disease symptoms vary with the infecting pathogen and the infected part and can include leaf spots, leaf blights, root rots, fruit rots, fruit spots, wilt, dieback and decline.

Worldwide, per capita availability of food is projected to increase around 7 percent between 1993 and 2020, from about 2,700 calories per person per day in 1993 to about 2,900 calories. This is gradually becoming a mere dream because of plant disease which reduces yields. This implies plant disease have both direct and indirect impact on health, food security and economic growth of every nation. Since plant diseases are strongly influenced by environmental factors, it will be unrealistic to talk about all plants and all diseases (they are heterogenous). Chile is one of the most popular and promising grown plant in New Mexico.

**Motivation**

New Mexico is the nation’s largest Chile pepper grower, followed by California, Arizona and Texas. It is obvious that Chile farm and produces are insufficient as about 80 percent of the Chile peppers consumed in the United States are imported, largely due to lower hand labour costs, lack of adequate funds and disease control.

Accurate and early identification is essential in tracking plant disease. Initially, the identification of plant diseases solely relies on visual examination. The process is not efficient and is also prone to human error. For a trained computer with classifier algorithms, diagnosing plant disease becomes easy and efficient. Machine learning algorithms recognize plant disease type, severity, and so on by sorting through hundreds to thousands of photos of diseased plants (Samuel, 2017).

The goal of this project is to build a robust model and application using machine and deep learning approach for image classification of Chile pepper plant disease. This will be achieved using keras package with python (by Francois Chollet and J.J. Allaire, 2018) with TensorFlow (by google, 2015 and updated January 2018) as backend. The main objectives are to use three techniques to address overfitting and train a robust model.

**Overfitting:** Overfitting is one of the big problems of both machine and deep learning models especially in the cases of limited training data samples. It occur when model performance is very high on training dataset and very low on either validation, test or both. This study tackle the problem by implementing the technique of image augmentation and dropout regularization. Biasness was addressed by using three way random data-splitting method (Training, Validation, and Testing), 13% of the whole data set removed entirely and stored in a different folder for final test purposes. The remaining were later split into 70-30% training–testing part.

**Methods**

**Data set Description.**

The considered pathogenic disease affect Chile plant stem and thereby block both water and nutrient intake for leaves. This study recognizes area base effect on intensity values of images, thereby uses images peculiar to the area of New Mexico state. 3002 images were collected between May and August of the year 2018 and 2019 from three different areas of New Mexico state (Las Cruces, Deming, and Los Lunas). Each categories has equal values of 1501. The images with two class labels namely “Disease” and “Normal” were analysed with CNN using python 3.6 software.

**Prepossessing**

Adequate model performance solely relies on whether the data is formatted into appropriate floating tensor points before feeding it into the network. The JPEG picture files was read in and decoded to RGB grids of pixels. Images were resized to 50 x 50 pixels and model normalization, optimization and predictions were performed on these downscaled plant images. Rescaled pixel values from 0 - 255 to 0 – 1. Keras package handles some prepossessing, batch size = 20

**Deep Neural Network (DNN)** is an Artificial Neural Network (ANN) with multiple hidden layers between input and output layers. It can be supervised, partially supervised, or unsupervised. The supervised deep learning was considered in this study.

**Loss Function**

The robustness of any machine or deep learning model depends on how low the value of the loss function is. It is a very important part of artificial neural network modelling. It is used to measure how close is the predicted response to actual labelled response y. the smaller the value of loss functions the better the model.

The loss function is given as

**Gradient Descent**

Gradient descent is essential to find the optimum value of that minimize the loss function (). The approach is the first order optimization process which is used to find local minima of an objective function.

Gradient descent can be mathematically represented as follows.

Is the number of repetitions until convergence is reached. learning rate

The main problem with this traditional gradient descent approach is that it requires long training time large dataset. This problem can be overcome by considering stochastic gradient descent (SGD). SGD requires having dataset in batches . The mathematical representation of SGD is given as follows

k is the number of batches.

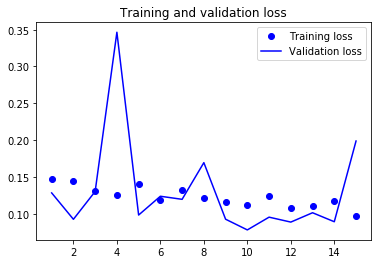
**Image augmentation** technique was implemented by artificially expanding dataset. The parameter used were rotation, zoom, shear and preprocessing functions. In addition, augmented images were generated by custom function for contrast stretching, histogram equalization and adaptive histogram equalization.

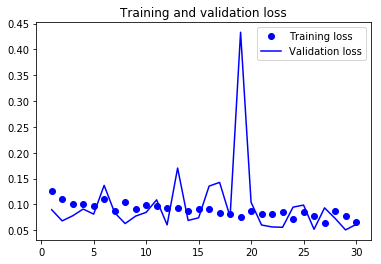
This research, the rotation range was set to be 40 degrees. The flip is horizontal with Width shift, height shift, shear range, and zoom range were all set to 0.2.

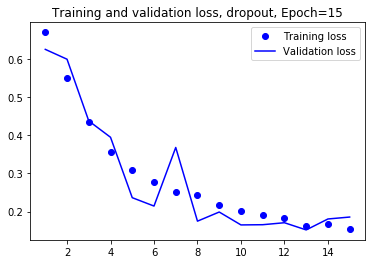
**Results**

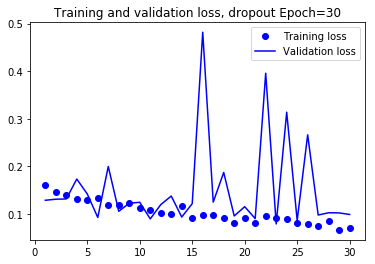
**Accuracy Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Training Accuracy** | **Validation Accuracy and F1-Score** | | | |
|  | Epoch |  | Accuracy (%) | Class accuracy (%) | | F1-Score (%) |
|  |  |  |  | ***Disease*** | ***Normal*** |  |
| **CNN model** | 15 | 95.07 | 91.87 | 82.60 | 100 | 90.47 |
| 30 | 97.54 | 98.02 | 97.30 | 99.73 | 98.51 |
| **CNN (Dropout)** | 15 | 93.36 | 92.70 | 97.30 | 90.35 | 91.01 |
| 30 | 97.63 | 98.19 | 99.02 | 98.93 | 98.97 |

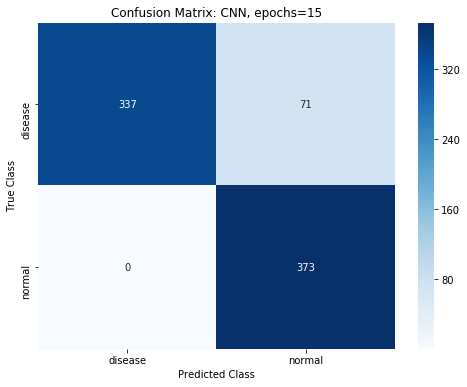


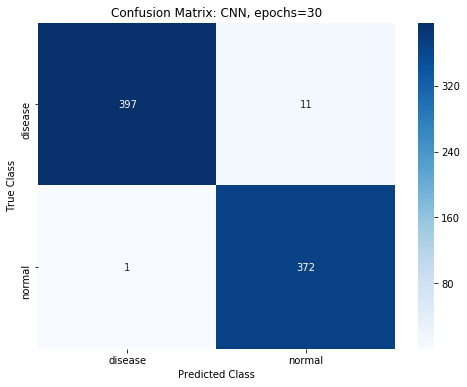


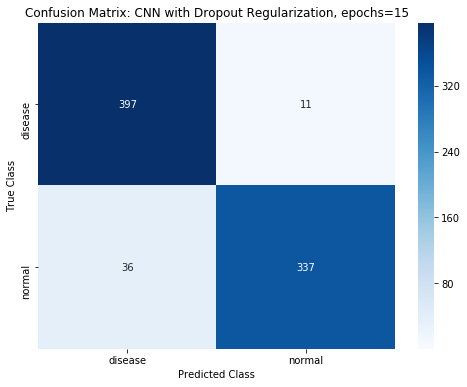
****

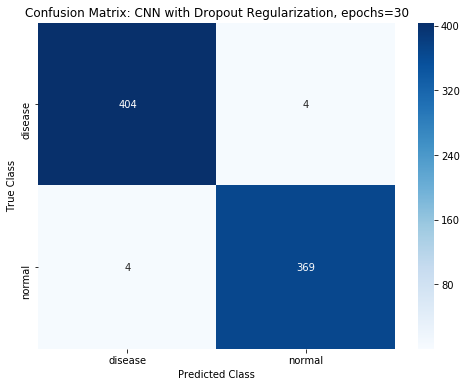
****

**Confusion matrix**

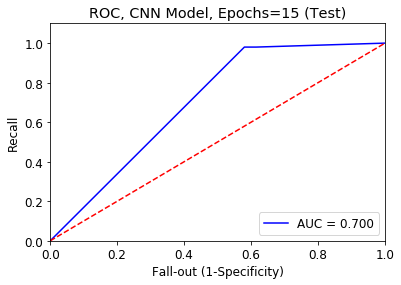
****

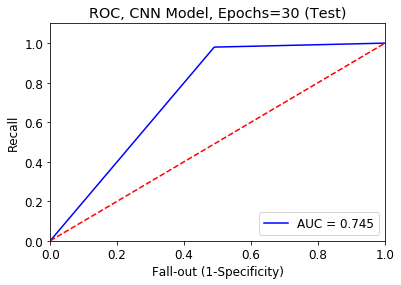
****

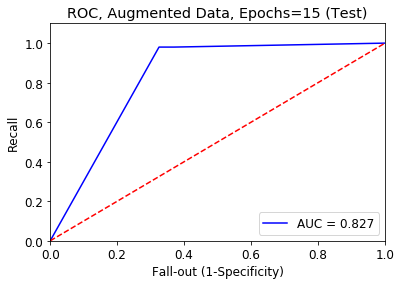
****

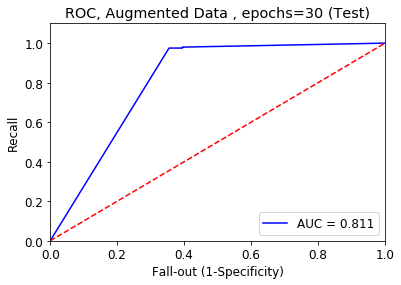
****

**ROC and AUC curve**

****

****

****

****