

**Computer Vision Semester Project**

**Group Members:**

**Name:** Hassan Mahmood

**Roll No:** SP20-BCS-114

**Email:** [SP20-BCS-114@cuilahore.edu.pk](mailto:SP20-BCS-114@cuilahore.edu.pk)

**Phone No:** +92309-4644191

**Name:** Muhammad Talha

**Roll No:** FA20-BCS-007

**Email:** [FA20-BCS-007@cuilahore.edu.pk](mailto:FA20-BCS-007@cuilahore.edu.pk)

**Phone No:** +92313-0600459

**Potato Disease Classification**

**Abstract:**

The agricultural sector faces significant economic losses annually due to diseases affecting **potato crops**, particularly Early Blight and Late Blight. Identifying these diseases early is crucial for **farmers** to implement timely treatments and prevent substantial economic losses. In response to this challenge, we propose the development of **a computer vision-based application**, accessible through web platforms, for accurate detection of potato plant diseases.

The application is designed to accept images of **potato plant leaves** captured through a device's **camera**. These images are processed by the trained model, which accurately classifies the health status of the plant into one of three categories: Healthy, Early Blight, or Late Blight. The **prediction** includes the class and its associated **probability**, providing users with valuable information for informed decision-making.

**Introduction:**

Agriculture, backbone of our **global economy**, is consistently challenged by the impact of crop diseases, resulting in substantial economic losses for f**armers**. Among the critical crops affected, **potatoes** face significant threats from diseases such as **Early Blight** and **Late Blight**. The timely detection and effective management of these diseases are crucial for sustaining crop health and securing agricultural yields.

This **project** will help farmers by leveraging cutting-edge technologies in computer vision and machine learning. Our objective is to develop an **intelligent application**, accessible via platforms, capable of accurately **identifying and classifying** diseases affecting potato plants. Through the use of **deep learning models**, specifically Convolutional Neural Networks **(CNNs),** our solution aims to provide farmers with a reliable tool for **early disease detection**.

The **project** begins by acquiring **Potato plant leaves images** as need to train neural network on dataset, so using a well annotated dataset from **Kaggle**. Source provided below

<https://www.kaggle.com/datasets/emmarex/plantdisease>

Dataset have total 2152 images in which 152 images for Healthy leaves class and **1000 images** for **early blight** and **late blight** respectively. All images are in **RGB** and of **256x256x3**. Dataset undergoes **data preprocessing** steps to ensure its quality and diversity. To enhance the **model's learning** capabilities, **data augmentation** techniques are applied to the dataset such as **random Rotate** and **random flip**. The convolutional neural network (**CNN**) model is then **trained** on this **prepared dataset**.

In addition to the model development, the project emphasizes the seamless integration of the trained model into an **application environment**. The backend of the application is implemented using **FLASK**, allowing for the creation of a server side for model deployment. As, frontend of app we used **html with bootstrap**. Our application take an image from user and then process this image and resize it using open-cv library, then Using **potatoes.h5** file which carries parameters of model, we predict the class of image and display to user on web application.

**Architecture and Methodology**:

We will be developing an Application **WEB** which involves **Computer Vision** for detection of each disease accurately by having a picture of leaf of potato plant using camera and passing it to application. There are **3 Classes** whichour model will be predicting **Healthy**, **Early Blight** and **Late Blight**. So, our app will show the Predicted Class with probability.

This project uses **Tensor Flow’s Keras** library for comprehensive **data processing, augmentation, and training**. The dataset obtained from Kaggle, is **pre-cleaned, annotated**, and comprises images consistently **sized** at 256 x 256 x 3 pixels. So, not much processing required on the dataset as already dataset is very clean.

**Data Augmentation:**

To diversify the dataset and enhance the model's generalization, advanced data augmentation techniques such as **Random Flip** and **Random Rotation** are applied.

**Data Splitting:**

Firstly, we divided our images dataset into **32 sized batches** so total we have **68 batches** which carries our images as it will help to use **memory efficiently** while training model, so our training time will reduce. Then dataset is intelligently **shuffle** first as we do not want to train our model which can learn patterns. After that data is split into **training (80%),** **validation (10%),** and **testing (10%)** sets, ensuring a representative distribution of classes across the splits. So, total **56 batches for training** and **6 batches** for **validation** and **testing** each.

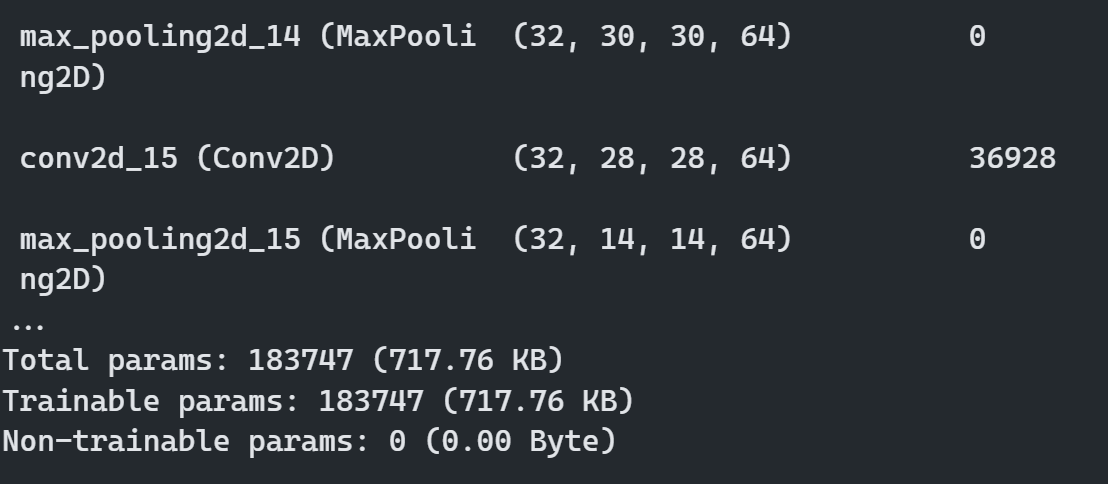
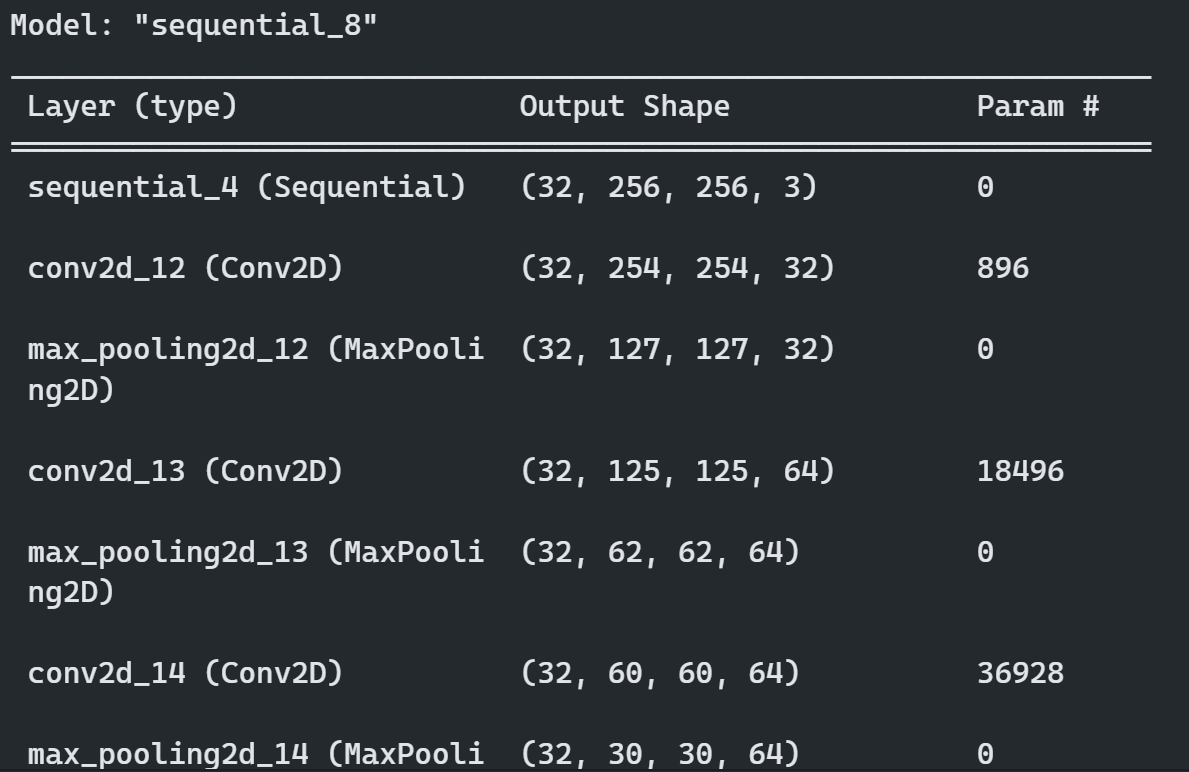
**Approach:**

Once data is ready, we have **two approaches** to train our model. One is using traditional **machine learning classification algorithms** like **SVM** and **KNN** with **feature extractor** and **disruptor** like **HOG** and **SIFT.** Second is using Convolution Neural Network (CNN), beauty of this method is that it extract features by its own and results are way much better compare to first method. So, I uses **CNN** as our Model with **Adam** as a parameter Optimizer in backpropagation.

**CNN:**

As uses Convolutional Neural Network (CNN) with **six layers**, each having **a 3 x 3 filter** which is also called as **Kernel Size** and each layer is followed by **max-pooling layers** with **2 x 2** convolving **window**. **ReLU** (Rectified Linear Unit) activation function is used in each **convolution Layers** which are strategically placed to capture **hierarchical features** in the data. Once image will be pass through these 6 layers of convolution then there multiple **fully connected layers (FC)** which are hidden layers of neural network also use ReLU as its activation function.

I uses **32 filter** of 3x3 size at first layer of Convolution and then onward uses 64 filters in remaining 5 layer of Convolution in network. Summary of CNN model is given below



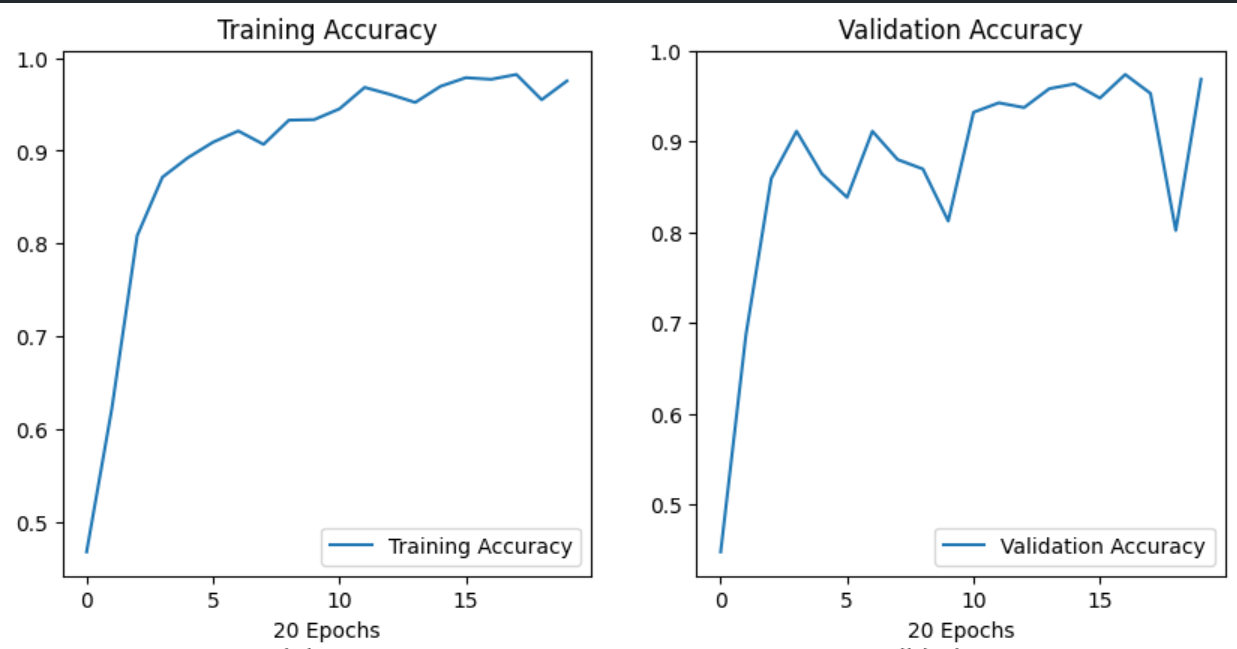
**Epochs:**

Total Epochs for training the model of dataset, I set is **20** as model is learning very well. For training the model, total **183747 parameter** model need to learn which are weight and biases of different layers.

**Model Training:**

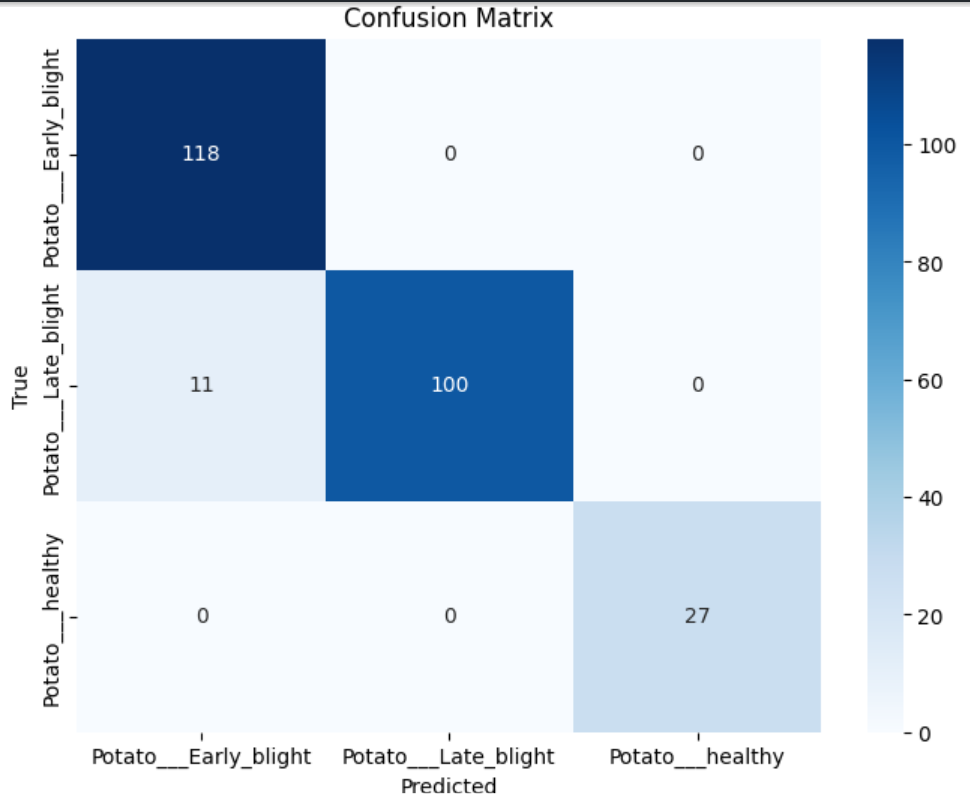
Model is trained on **Batches** of images which carry shuffled images of different classes. So, Model is train in 20 Epochs as I mentioned early. While Training, model is learning very fast and **Adam** is uses as Optimizer so, it is updating parameter very efficiently, so only **30-45 mins** it take to learn of the dataset at our local machine. For Lose, I use Cross Entropy and it’s calculated at each iteration. After training the model, I **save** the model in **potatoes.h5** file so we can use its parameter so real-time classifications. Once, my model is Trained, My accuracy is around **96 percent** for **training** and **validation. Lose** is around **0.13**, which is very impressive.

So, further loss graph and confusion matrix is shown

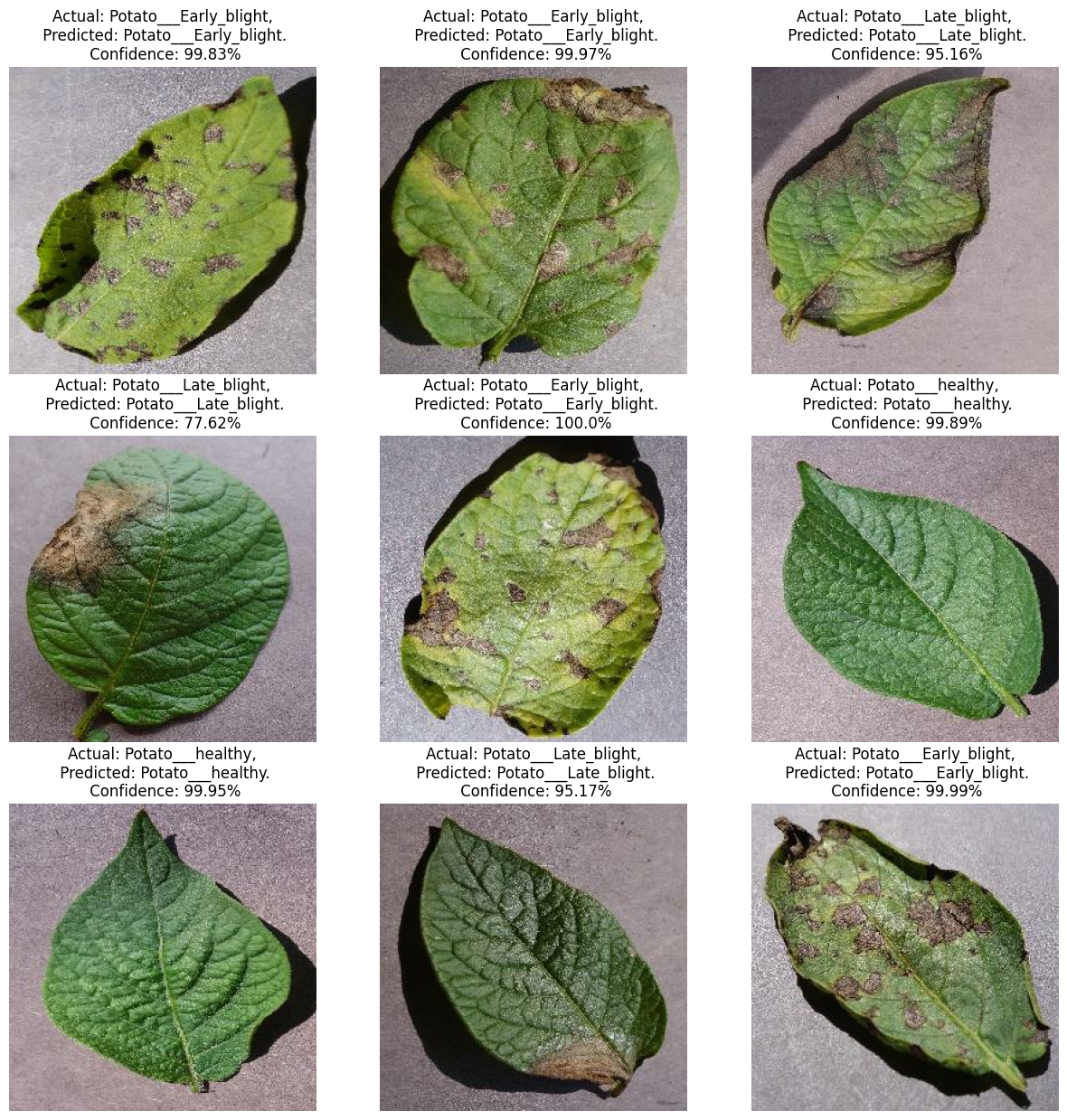




**Confusion Matrix:**



**Our Results:**



**Our Web Application:**



