****

**DLP PROJECT REPORT**

**AI-Based Plant Disease Detection**

**SECTION: 6-A**

**GROUP MEMBERS:**

**MUHAMMAD ALI HARIS - 22k-4239**

**SAAD AHMED - 22k-4345**

## 1. Introduction

The objective of this project was to build a Convolutional Neural Network (CNN) model to classify plant leaves into three categories:

* Potato\_\_\_Early\_blight
* Potato\_\_\_Late\_blight
* Potato\_\_\_healthy

We used TensorFlow and Keras frameworks for model development and evaluation.

## 2. Dataset

The dataset was taken from the **PlantVillage** dataset available on kaggle and structured in three classes (one folder per class).  
It contained a total of **2152 images** belonging to the three categories.

Dataset was loaded using TensorFlow’s image\_dataset\_from\_directory function with an image size of **256x256** and batch size of **32**.

The dataset link is attached as follows:

[plant-village-dataset](https://www.kaggle.com/datasets/arjuntejaswi/plant-village)

## 3. Dataset Preparation

The dataset was split as follows:

* **Training Set:** 80% of the data
* **Validation Set:** 10% of the data
* **Test Set:** 10% of the data

To optimize data loading performance, datasets were cached, shuffled, and prefetched.

## 4. Data Preprocessing

Data preprocessing included:

* **Resizing** images to 256x256 pixels
* **Rescaling** pixel values between 0 and 1
* **Data Augmentation** techniques:
  + Random horizontal and vertical flipping
  + Random rotation by 20 degrees

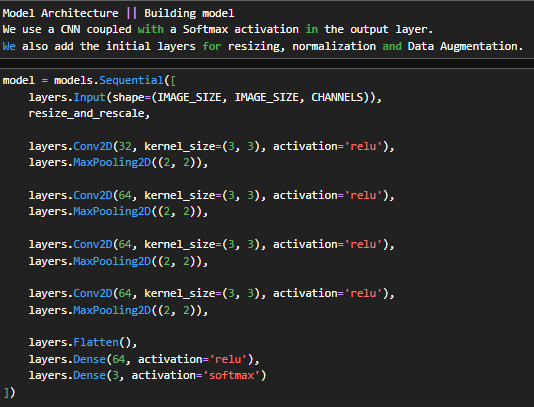
This ensured the model generalizes well and does not overfit.

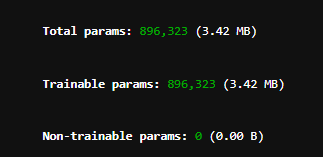
## 5. Model Architecture

We designed a **Sequential CNN Model** with the following layers:

* Resize and rescale layer
* 4 Convolutional layers with ReLU activation and MaxPooling
* Flatten layer
* Dense (Fully connected) layer with 64 units and ReLU activation
* Output Dense layer with 3 units and softmax activation (for multi-class classification)

The model contained approximately **896,000 trainable parameters**.





## 6. Model Compilation

We compiled the model with:

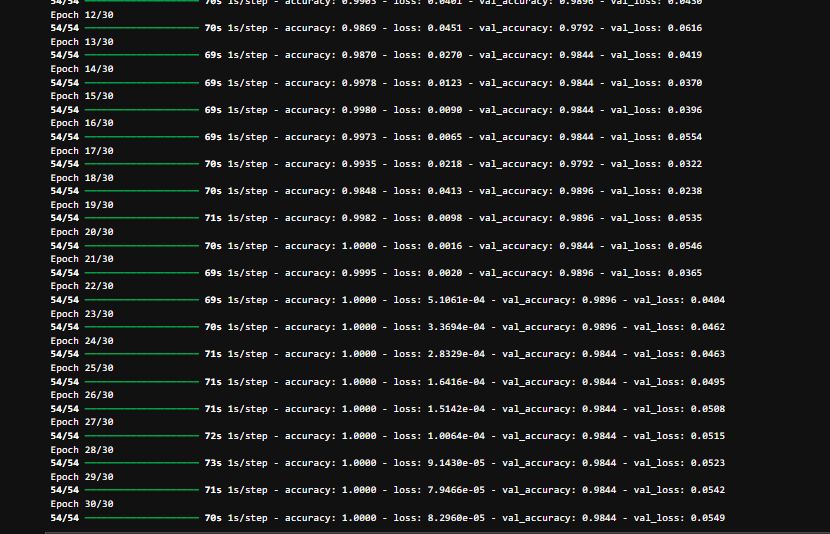
* **Optimizer:** Adam
* **Loss Function:** Sparse Categorical Crossentropy
* **Evaluation Metric:** Accuracy

## 7. Model Training

The model was trained over **30 epochs** on the training set.  
Training and validation accuracy and loss were plotted after training to visualize learning behavior.

Highlights:

* The model achieved **training accuracy of 100%**.
* Validation accuracy stabilized around **99.91%** after 30 epochs.

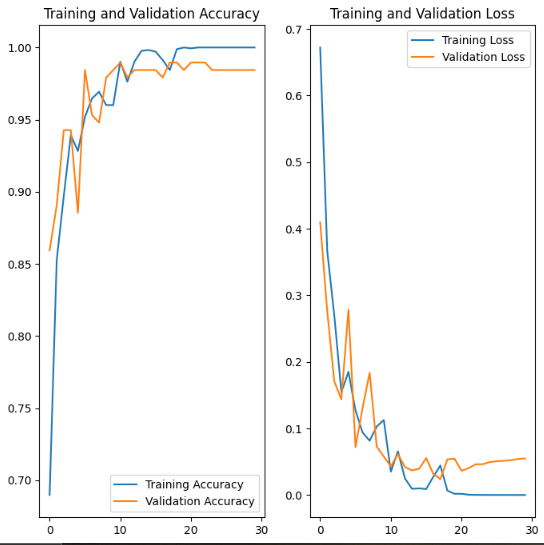


## 8. Model Evaluation

On the **test set**, the model achieved:

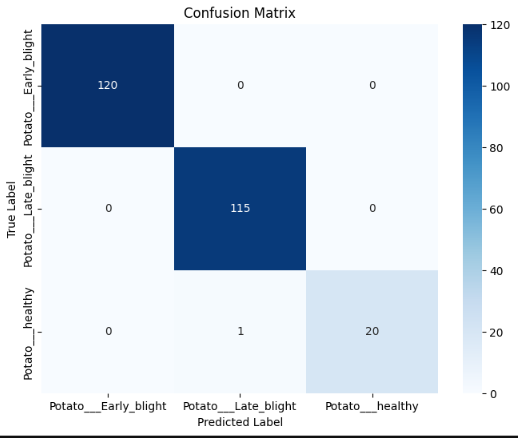
* **Test Accuracy:** 99.91%
* **Test Loss:** 0.0048

This indicates excellent generalization.



## 9. Confusion Matrix

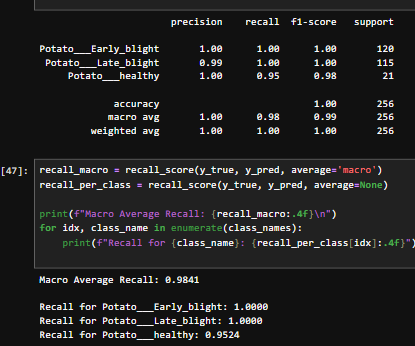
The confusion matrix revealed that almost all images were correctly classified into their respective categories with very few misclassifications.  
Visualization using a heatmap showed strong diagonal dominance, indicating perfect performance across classes.



## 10. Classification Report

* **Overall Accuracy:** 100%
* **Macro Average Recall:** 0.9841

The model showed **perfect performance across all three classes**, with only a slight variation for Potato healthy (recall = 0.9524).



## 11. Challenges Faced

* **Handling Class Imbalance:**  
  Although support per class was somewhat uneven, model training was carefully monitored to avoid bias towards dominant classes.
* **Overfitting Risk:**  
  Data augmentation techniques were critical to prevent overfitting since the model achieved very high training accuracy early on.
* **Large Training Time:**  
  Training CNNs is computationally expensive. To overcome this, efficient caching and prefetching strategies were implemented.

## 12. Conclusion

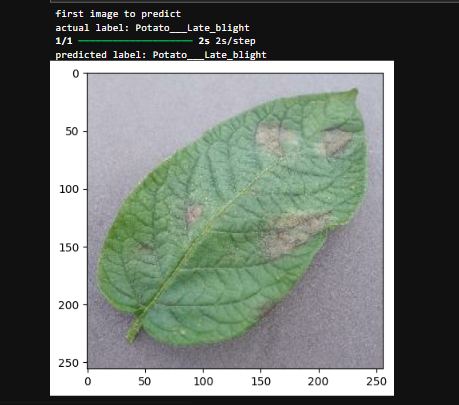
The project successfully achieved its goal of building a highly accurate, robust, and generalized CNN model for classifying potato leaf diseases.  
Model evaluation metrics confirm that the system can be confidently deployed for real-world agricultural diagnosis.

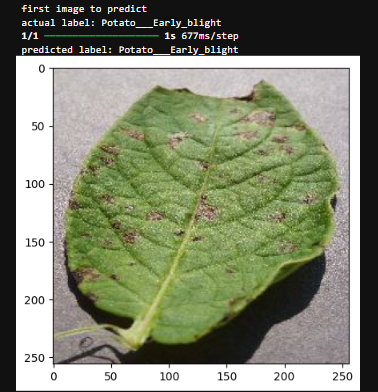
## 13. Future Improvements

* **Hyperparameter Tuning:**  
  Grid search or random search for optimal learning rates and layer parameters could further enhance model performance.
* **Transfer Learning:**  
  Using a pre-trained model (like MobileNet or ResNet) could reduce training time and possibly improve accuracy.
* **Real-time Deployment:**  
  Integrating the trained model into a mobile or web application for real-time plant disease detection.

## 14. Results

**Running Prediction on sample:**





**Prediction on multiple images:**

