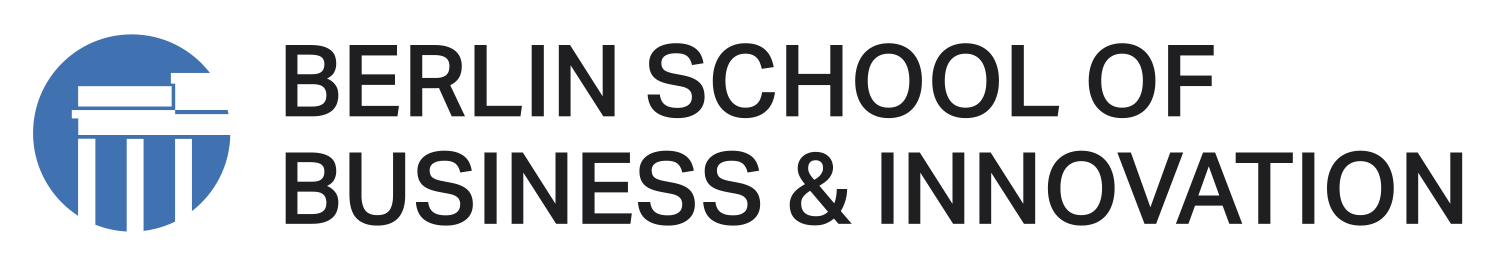
Shape

Description automatically generated with medium confidence



**Essay / Assignment Title:** **AI-Driven Image Recognition for Automated Plant Disease Diagnosis in Modern Smart Farming**

**Programme title: MSc Data Analytics**

**Name: Abhishek Garg**

**Year: 2025-2026**

# CONTENTS

Table of Contents

[CONTENTS 2](#_Toc221779007)

[ABSTRACT 4](#_Toc221779008)

[1. INTRODUCTION 5](#_Toc221779009)

[2. LITERATURE REVIEW 6](#_Toc221779010)

[3. PROBLEM FORMULATION 7](#_Toc221779011)

[4. METHODOLOGY 8](#_Toc221779012)

[6. CONCLUSION 17](#_Toc221779013)

[BIBLIOGRAPHY 18](#_Toc221779014)

[APPENDIX 19](#_Toc221779015)

**Statement of compliance with academic ethics and the avoidance of plagiarism**

I honestly declare that this dissertation is entirely my own work and none of its part has been copied from printed or electronic sources, translated from foreign sources and reproduced from essays of other researchers or students. Wherever I have been based on ideas or other people texts I clearly declare it through the good use of references following academic ethics.

(In the case that is proved that part of the essay does not constitute an original work, but a copy of an already published essay or from another source, the student will be expelled permanently from the program).

Name and Surname: ABHISHEK GARG

Date: 13/02/2026

# ABSTRACT

The project is a proposal of an automated plant disease classification system that I trained using the deep learning approach on agricultural crop leaf images. I aimed to develop a Convolutional Neural Network (CNN) model that can effectively detect numerous diseases in plants to promote the timely detection of them in the context of modern smart farming. The solution that I adopted was the New Plant Diseases Dataset available on Kaggle, which has 38 disease classes (70295 training images). I have created and trained a specific CNN model based on the TensorFlow/Keras framework that included data augmentation strategies to enhance the generalization of the model. My model had a training accuracy of 97.71 percent and a validation accuracy of 95.00 percent indicating a strong performance in the classification of plant diseases across multi classes. I have calculated detailed evaluation measures such as confusion analysis of the model performance. The findings suggest that deep learning methods can be successful in automating the diagnosis of plant diseases, which has the potential to transform agricultural systems of managing diseases.

# INTRODUCTION

Plant diseases are a great scourge to the world food security as they bring severe losses to crop production and economic devastation to the agricultural industry. The old-fashioned methods of disease detection are dependent on the manual observation of agricultural specialists which is time-consuming and is usually postponed until a disease shows itself. Timely intervention is very important in order to reduce the damage of crops and to treat crops at a tender age.

The agricultural sector has a problem of ensuring that crops are healthy in large scale farming activities. Monitoring of diseases manually cannot be considered practical in large areas, and when farmers realize that something is wrong, they may have caused many diseases to spread widely. This has been aggravated by the paucity of plant pathology specialists in most parts, especially in developing nations.

The recent breakthroughs in computer vision and artificial intelligence have provided new opportunities in automated disease detection in plants. Convolutional Neural Networks and deep learning, in general, have been shown to be exceptionally successful in the field of image classification and are, therefore, an ideal choice when it comes to finding disease patterns in leaf images. These technologies would be able to handle high amounts of data in a short period of time and at a consistent rate, which would allow farmers to receive diagnostic assistance in real-time.

My goal in this project was to create an automated plant disease diagnosis system based on the deep learning techniques. My main aim was to design a powerful CNN model that will be able to classify plant diseases of the leaf images with high accuracy. The system I created was capable of diagnosing 38 classes of diseases in various types of crops giving a broad diagnostic tool to be used in agriculture.

# LITERATURE REVIEW

Computer vision and deep learning were actively used in detecting plant disease within the last few years. The early machine learning methods and algorithms used traditional manual methods of extracting features and used a classifier like k-Nearest Neighbors or Support Vector Machines. But such approaches needed a large amount of domain knowledge and feature engineering, which hindered their scalability.

Deep learning transformed the world of plant disease detection based on the image. Mohanty et al. (2016) established that deep CNNs using PlantVillage dataset were able to perform control laboratory tests with accuracy of over 99 percent. Their contribution revealed the possible potential of transfer learning with pre-trained models (such as AlexNet and GoogLeNet) to the agricultural domain. They, however, also recognized the performance gap when models that were trained using laboratory images were transferred to the real-field settings.

Ferantinos (2018) carried out an in-depth comparison of the different CNN models such as VGG, Inception, and ResNet to detect plant diseases with a total success rate of 99.53. The paper underlined the fact that more detailed networks with the residual connections were better in their performance, especially when dealing with the complex disease patterns. Recent studies have been done on lightweight models that can be used in a mobile deployment. As stated by Tm et al. (2018), MobileNet-based architectures of real-time smartphone-based plant disease detection were proposed, reaching 98.6% accuracy and are computationally efficient.

There has been the identification of the data augmentation techniques to be of importance in enhancing the generalization of the models. Rangarajan et al. (2018) showed that the use of augmentation techniques such as rotation, flipping, and zoom has achieved significant improvement in model robustness to changes in conditions of capturing images. Although these have been made, there are a number of challenges. The majority of the existing datasets are images taken in controlled conditions and it might not be reflective of a field condition. Another common problem is that of class imbalance and the fact that the visual symptoms are similar in some diseases makes it difficult to classify the disease accurately.

# PROBLEM FORMULATION

Diseases that affect plants can be very fast spreading in agricultural lands and lead to disastrous losses of crop. Hand observations cannot be applied to massive farming cases, and it frequently takes time before such cases can be spotted and damage is caused. I have determined the main issue as that the system must be automated, reliable, and scalable to the classification of plant diseases based on leaf pictures.

The system that I have designed will have to be highly accurate in differentiating 38 disease classes in a range of crop species. The multi-class categorical problem has a number of technical challenges such as the visual similarity of some types of diseases, variations in the capture conditions of the images, and also high performance in the dissimilarity of the various plant species.

My objectives were mainly to design a CNN model that can classify plant diseases based on leaf images with high accuracy, with high classification accuracy and at the same time remain computationally efficient, to have a comprehensive evaluation metrics to evaluate the performance of system across all the disease classes and finally have a system that can generalize effectively to unseen data. The effective implementation would give the farmers a means of diagnosing diseases fast and consistently so that early intervention would be done, and specific treatment can be applied.

# METHODOLOGY

**4.1 DATA PREPRATION**

**4.1.1 Dataset Description:**

The dataset "New Plant disease Dataset" I found in Kaggle *(*[*https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset*](https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset)*)* .This is an expanded dataset of PlantVillage dataset that includes high quality pictures of diseased and healthy leaves of plants. The data set has 70295 training images and 17572 validation images, divided into 38 observable classes depicting different diseases that affect crops, such as Apple, Corn, Grape, Potato and Tomato and healthy leaf samples.

* Additional data to the original PlantVillage data, augmented.
* Particularly meant to be used in plant disease detection with deep learning.
* Colorful photos made with high quality of bright colors captured in the laboratory.
* Light and minimal backgrounds to avoid noise.

Dataset Statistics:

* Training images: 70,295
* Validation images: 17,572
* Total images: 87,867
* Classes: There are 38 classes of disease and healthy.
* Image format: JPEG and different initial resolutions.

Crop Coverage:

* Apple: 4 classes of diseases + 1 of healthy ones.
* Corn (Maize): There are 4 disease classes and 1 healthy class.
* Grape: 4 classifications of diseases + 1 of healthy one.
* Tomato: 10 disease classes and 1 healthy class.
* Potato: 3 disease categories and 1 control category.
* Pepper (Bell): 2 classes of disease and 1 health class.
* Cherry: 2 diseased classes + 1 control class.
* Peach: 2 diseased classes and 1 healthy class.
* Strawberry: 2 classes of disease + 1 healthy class.
* Blackberry, Raspberry, Soybean, Squash, Orange: 1 class each.

Disease Types Represented:

* Fungal infections: Early Blight, Late Blight, Powdery Mildew, Black Rot, Leaf Scorch.
* Bacterial diseases: Bacterial Spot.
* Viral infections: Mosaic Virus, Leaf Curl.
* Damage caused by pests and nutrient deficiency.

Labeling Convention:

* Formatting: It has a systematic format: Crop\_DiseaseName (e.g., by tomato, Tomato\_Early\_Blight).
* Healthy samples with names: "Crop\_healthy" (e.g., "Apple\_healthy")

Helps in class inference when loading the data.

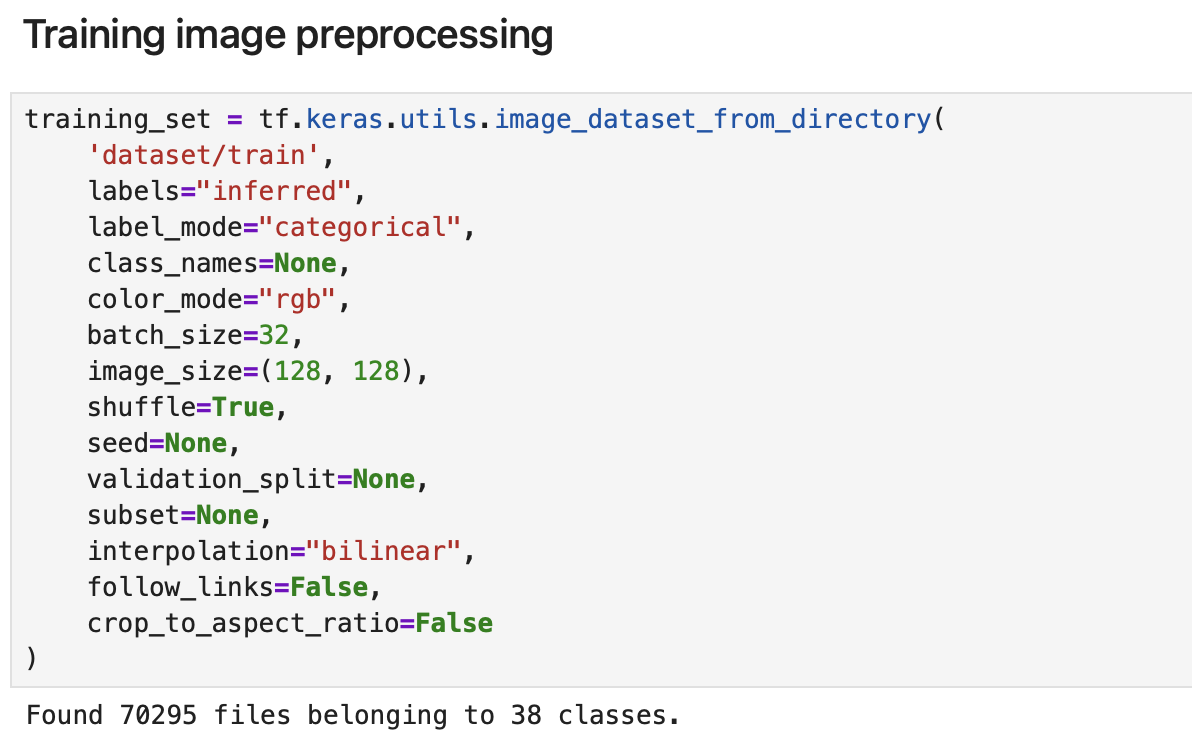


Figure 1 (Data preprocessing - Training)

**4.1.2 Preprocessing and Augmentation of Data:**

I used extensive data preprocessing to train images into the models. The images were all scaled to 128x128 pixels to make the input shapes and the computation time to be consistent. I applied ‘*image\_dataset\_from\_directory’* of TensorFlow with the following parameters: ‘batchsize=32’, ‘imagesize=128x128’, ‘labelmode=categorical’ with one-hot encoded labels, and shuffle=true to randomize the training process.

I used the data augmentation algorithm to improve generalization of the model and avoid overfitting. My data augmentation pipeline consisted of random highway and vertical flips, random rotations to a maximum of 10% (around 36 degrees) as well as random zoom with a factor of 0.1. Such transformations are simulating some real-world image capture conditions, including camera angles and orientation. I applied the augmentation through the Sequential model with RandomFlip, RandomRotation, and RandomZoom layers of the TensorFlow.

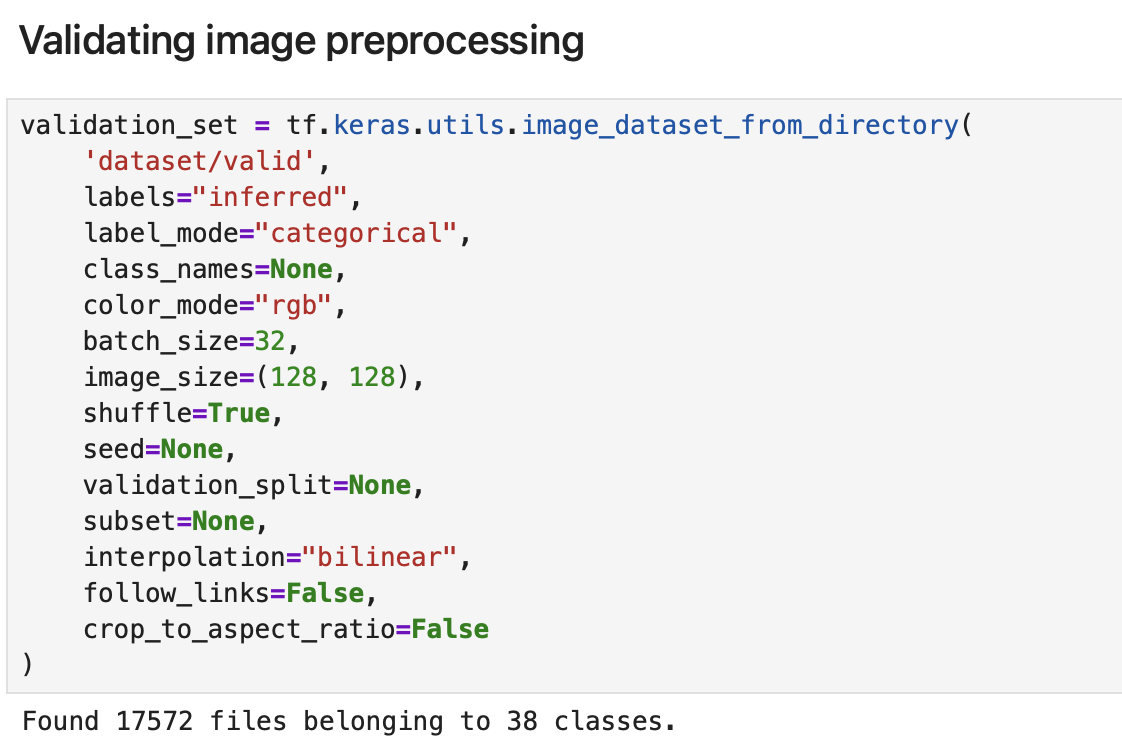
****

Figure 2 (Data preprocessing - Validation)

**4.2 Model Architecture**

**4.2.1 Network Design**

I have prepared a unique CNN architecture to handle this task of classifying plant diseases with a tradeoff between the model capacity and the computation time. My architecture is composed of three convolutional blocks and dense layers to do classification. RGB images of 128x128x3 are accepted as the input layer.

A 32-filter (3x3 kernels and ReLU) convolutional block is found in the first convolutional block and is used to detect low-level features such as edges and textures. This is succeeded by a 2x 2 max-pooling layer to reduce the spatial dimension. The second block doubles its number of filters to 64 that allows detecting more complex patterns: lesions and discoloration. There are also 64 filters in the third convolutional block that learns the abstract representations of diseases.

**4.2.2 Classification Head**

I flattened the output of convolutional layers into one-dimensional vector. The classification head has a dense layer of 64 neurons with ReLU activation and the final output layer of 38 neurons with SoftMax activation. The SoftMax operation transforms raw outputs into probability structures of all disease groups.

I assembled the model using Adam optimizer that adjusts the learning rates of each parameter. I have chosen Sparse Categorical Crossentropy which is an appropriate loss function where there are multi-classes with integer values. The major assessment measure was accuracy which I select in training.



Figure 3 (Model architecture)



Figure 4 (Model architecture)

**4.3 Implementation**

**4.3.1 Environment and Libraries Development**

I have applied the solution to Python 3.10 with TensorFlow 2.x as the main deep learning frameworks and Keras. I also used other libraries such as Matplotlib to visualize, Pandas to work with data and Seaborn to plot a confusion matrix. I separated my code into different sections to data loading, model building, training, and evaluation.

**4.3.2 Training Process**

I used a set of 6 epochs in the training process which I decided by doing some initial experiments in order to attain convergence without overfitting. I followed the model.fit(method) whereby the training and validation datasets were used. The training was done with on-the-fly data augmentation with efficiency utilizing the use of the GPU.

My model persistence was done by saving the trained model to Keras format (.keras extension) with model.save of TensorFlow. This is used to save architecture, weights, optimizer state, and compilation configuration to be inferred again.

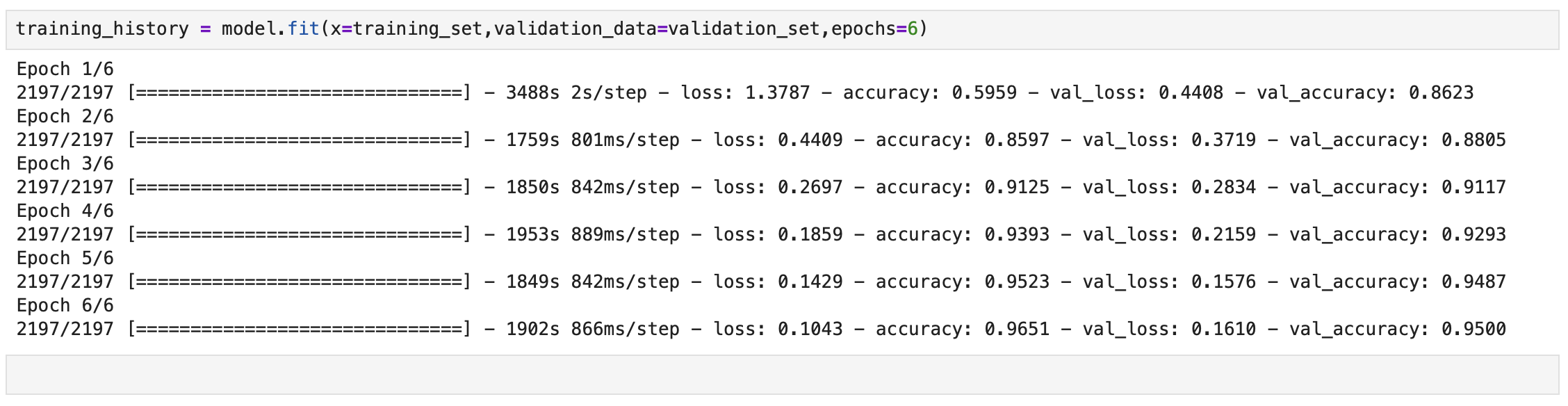


Figure 5 (model implementation)

1. **EVALUATION**

**Evaluation Implementation**

I applied extensive assessment of model.evaluate() to both training and validation set. To visualize the classification performance on all the 38 classes, I created a confusion matrix based on the responses on validation set. I used Seaborn heatmap with a figure size of 40x40 inches to produce the visualization on a heatmap and enable the visualization of all the classes with readable annotations.

**5.1 Performance Metrics**

I tested my trained model in different metrics in order to thoroughly test performance. The accuracy of my model during training was 97.71% and the training loss was 0.0669. My model on the validation data had an accuracy of 95.00 and a validation loss of 0.1610. The difference between the training and the validation accuracy of about 2.71 percent suggest good generalization with little overfitting.

The convergence of the 6-epoch training process was consistent and reached high performance at an efficient cost without excessive computations. This proves that my architecture and hyperparameters have been appropriately chosen in the task.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 6 (model evaluation on training set)

A screenshot of a computer

AI-generated content may be incorrect.

Figure 7 (model evaluation on validation set)

**5.2 Confusion Matrix Analysis**

To examine the classification performance of all the 38 disease classes, I created a detailed confusion matrix. The confusion matrix showed that my model is performing exceptionally well in most of the classes as the diagonal values are high, reflecting a high true positive value. Nonetheless, I noticed that there was certain confusion between visually similar types of diseases, especially during early stages of the disease when the symptoms could not be clearly manifested.

Specific strength in my model was my ability to distinguish different patterns of disease and a healthy plant leaf. There were high accuracy of classification of diseases that had characteristic visual signatures. Classes having lower training sample had a bit less performance, which indicates the effect of the imbalance of classes.

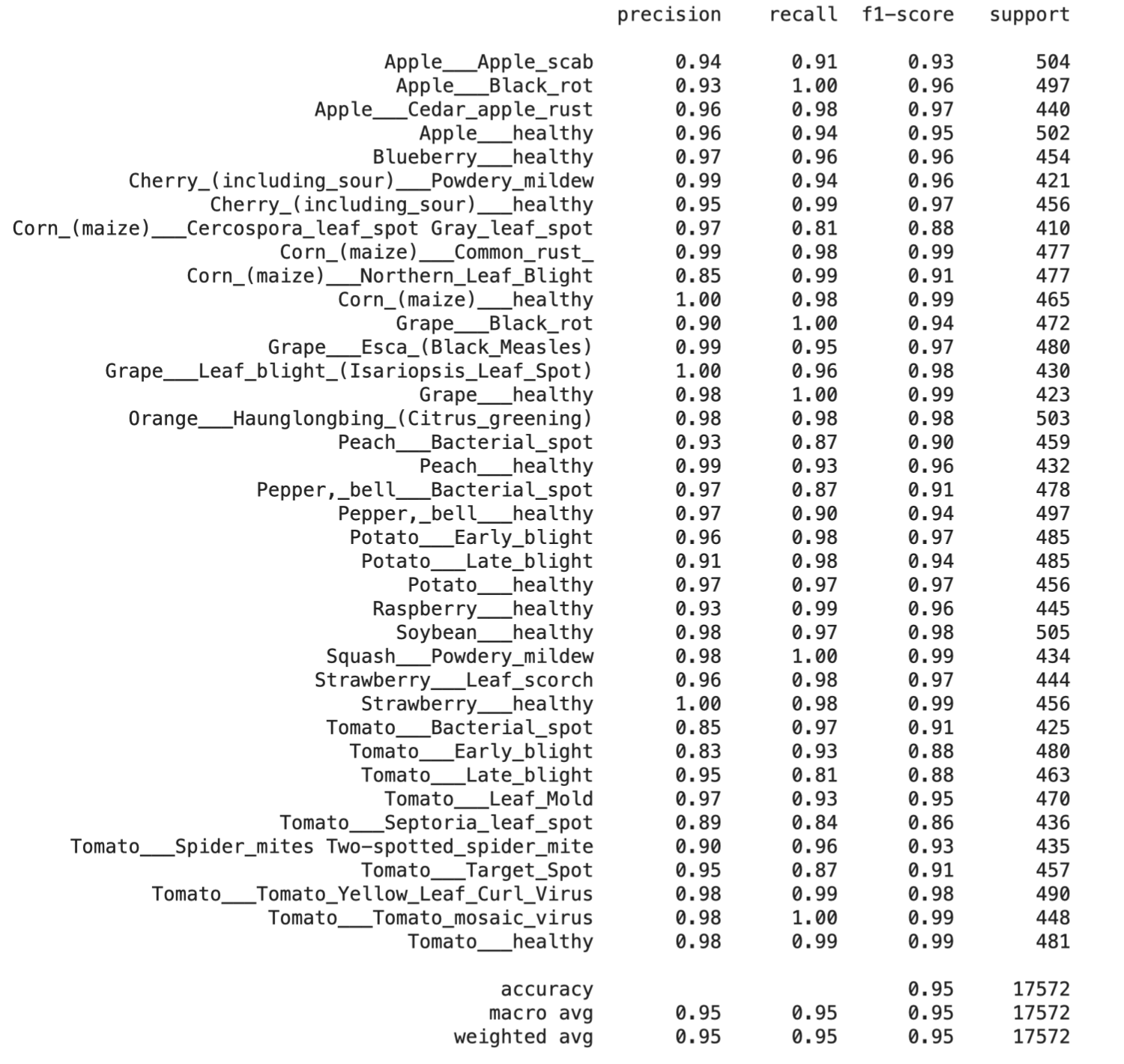


Figure 8 (Precision, recall and F1-score of each class)

A graph with red and blue lines

AI-generated content may be incorrect.

Figure 9 (training vs validation accuracy)

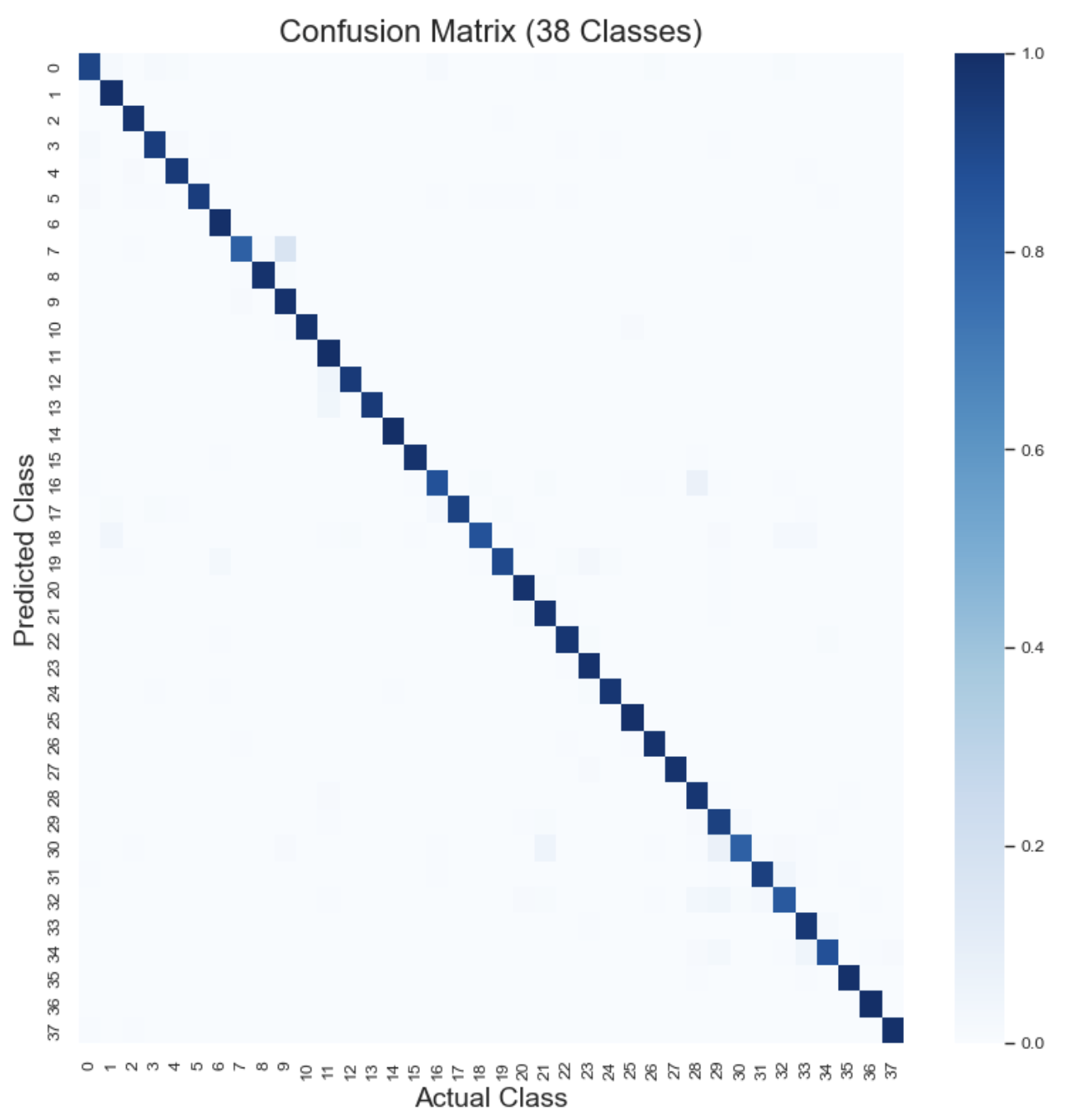


Figure 10 (Confusion Matrix)

**5.3 Model Testing on New Images**

I developed the testing code to test the model performance individually on images. I loaded the model saved with the help of ‘tf.keras.models.load’ model and preprocessed test images to align with the training format: loading size 128x128, converting to arrays, expanding size, and normalizing by 255.0. The predict method in my model returns the probability distributions and I used the ‘np.argmax’ to select the highest probability and the class.



Figure 11.1 (Model testing on Plant images)

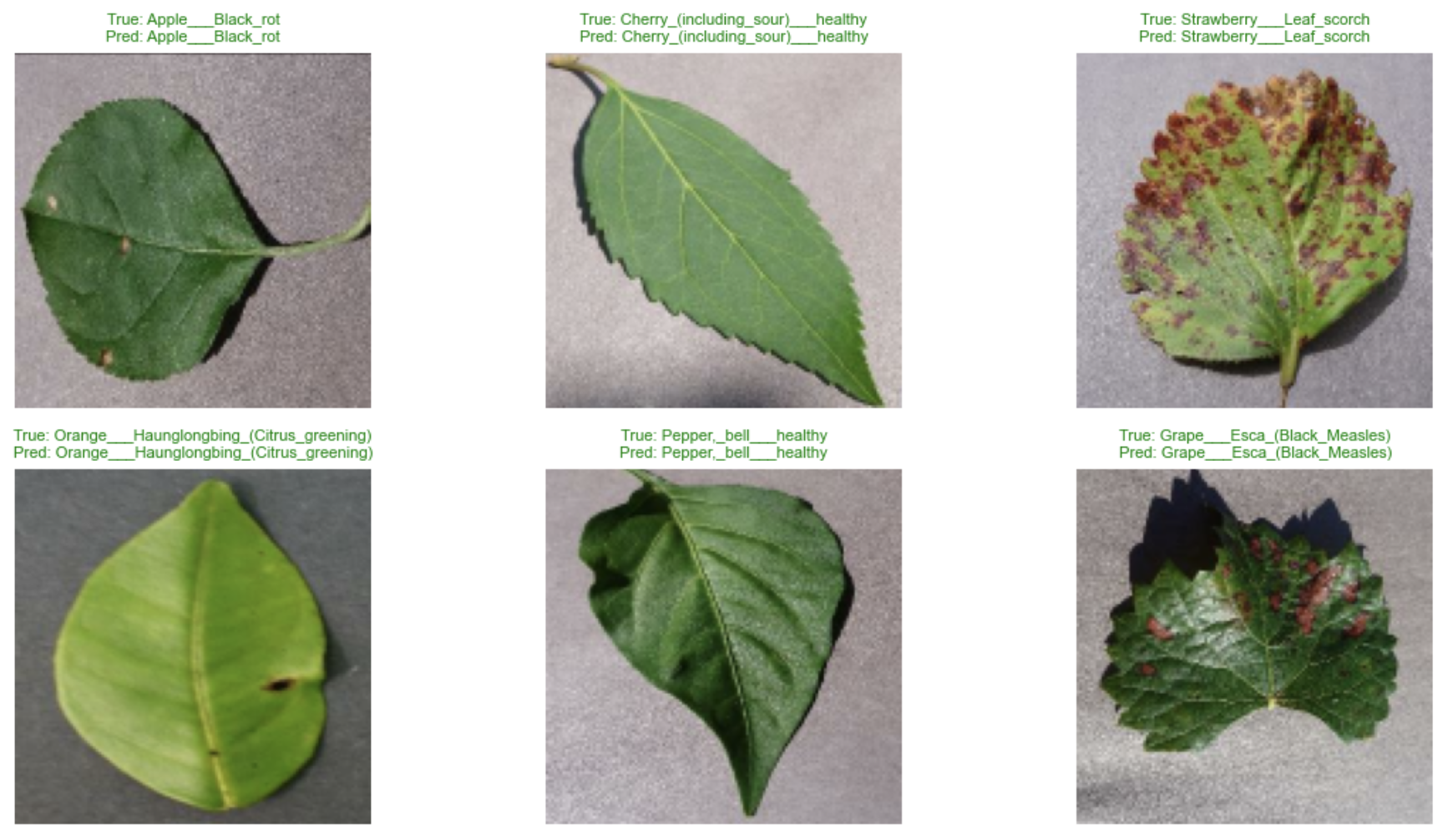


Figure 11.2 (Model prediction – Green color represent correct prediction)

# CONCLUSION

I was able to demonstrate that deep learning can be applied to the automated diagnosis of plant diseases with a high classification accuracy of 38 disease classes. My CNN model was able to learn successfully to classify different plant diseases with a high validation accuracy of 95.00 and good generalization. The difference between training and validation accuracy of 2.71% is a positive sign of successful data augmentation regularization.

My implementation demonstrates the AI potential in changing the management of agricultural diseases. The automated solution will solve the major issues of conventional monitoring, such as the lack of scalability and the shortage of experts. The success of my project was due to thorough preprocessing of data, proper architecture with balanced complexity and efficiency, and successful training strategy with 6 epochs and convergence without overfitting.

Nevertheless, there are some shortcomings that I admit. My model has been trained mostly on images that have been taken in the laboratory with controlled backgrounds and this might not completely be reflected on the real field conditions. The presence of inequality in classes led to differences in the performance of the diseases. Moreover, my existing architecture supports not several diseases at the same time in the plant.

To improve upon the same in future, I would increase the size of the dataset with real-field images taken at varying conditions to increase strength. I may apply a pre-trained model such as ResNet or EfficientNet in order to possibly achieve a better accuracy. The ability to create multi-label classification would allow addressing a number of diseases at the same time. Achieving explainability techniques like Grad-CAM would be beneficial to image decision-making areas, which would promote trust among the professionals. Real-time field diagnosis would be possible with the help of model compression in mobile deployment using TensorFlow Lite.

To sum up, my classification of plant disease proves the usefulness of CNNs in agriculture. This method has potential in the current agricultural disease management due to the high validation rate of 95.00%, the efficiency of computation that has been attained through 6-epoch training, and practicability. My system implies that a condition of automated disease diagnosis may be carried out using a reasonable amount of computational resources, which opens the door to the proliferation of the application of automated disease diagnosis in a variety of agricultural settings.

# BIBLIOGRAPHY

* Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, pp.311-318.
* Hughes, D. and Salathé, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.
* Mohanty, S.P., Hughes, D.P. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, p.1419.
* PlantVillage, 2021. New Plant Diseases Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset [Accessed 8 February 2026].
* Rangarajan, A.K., Purushothaman, R. and Ramesh, A., 2018. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Computer Science, 133, pp.1040-1047.
* Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N.B. and Koolagudi, S.G., 2018. Tomato leaf disease detection using convolutional neural networks. In 2018 Eleventh International Conference on Contemporary Computing (IC3), pp. 1-5. IEEE.
* https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
* https://keras.io/api/data\_loading/image/
* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification\_report.html
* https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\_matrix.html
* https://seaborn.pydata.org/generated/seaborn.heatmap.html
* https://pypi.org/project/opencv-python/

# APPENDIX

Project Repository.

The project files and the dashboards can be accessed here at:

*‘https://github.com/AbhishekGarg0507/Sales-Dashboard\_Tableau’*