**Plant Diseases Detection from Images Using CNN Model**

**Project**

Submitted by-

VenkataSubramanian S

MD112

**Introduction:**

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images. They have become the backbone of many computer vision tasks due to their ability to automatically learn spatial hierarchies of features from input images. CNNs are particularly effective for tasks like image classification, object detection, and image segmentation.

**Objectives:**

To Develop a comprehensive Streamlit application that allows users to upload images of plant leaves and accurately predict the presence and type of plant disease using a Convolutional Neural Network (CNN) model. This project involves designing, implementing, and optimizing a solution that integrates machine learning, computer vision, and user interface development

**Libraries Used:**

* Pandas
* Numpy
* Seaborn
* Matplotlib
* tensorflow

**Datasets:**

Link for Dataset: [New Plant Diseases Dataset](https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset)

Train Data:

* 70295 images belonging to 38 classes
* The size of the image is 128x128 and 3 channel.

Valid Data:

* 17572 images belonging to 38 classes
* The size of the image is 128x128 and 3 channel.

Test Data:

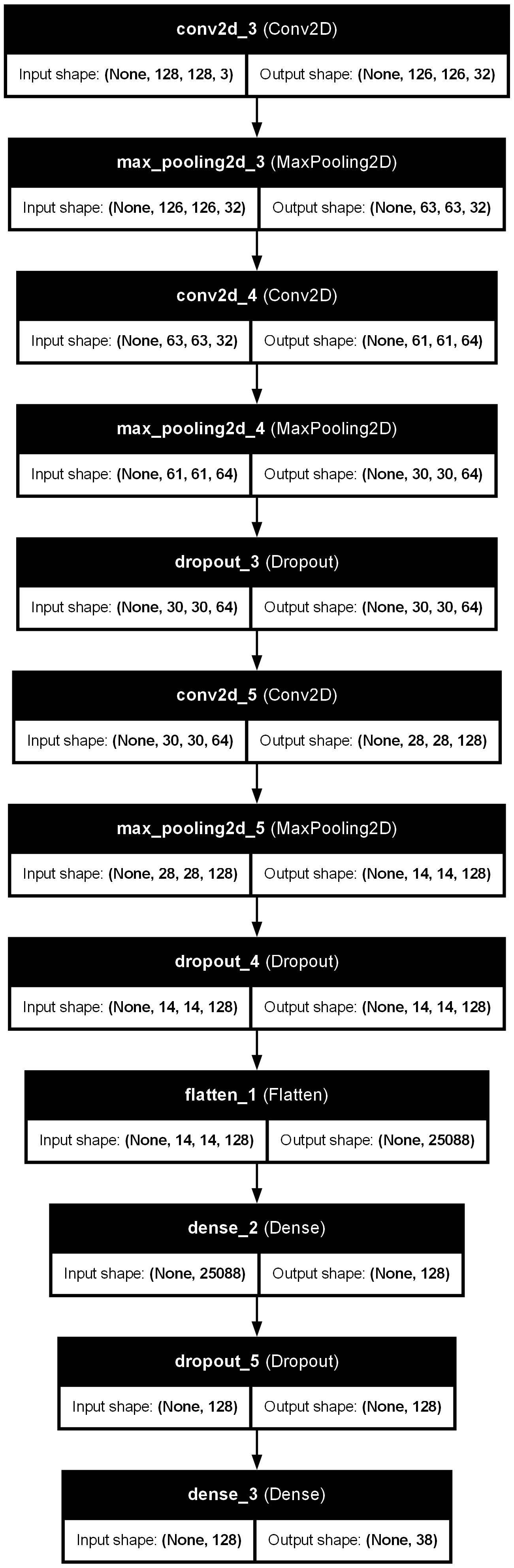
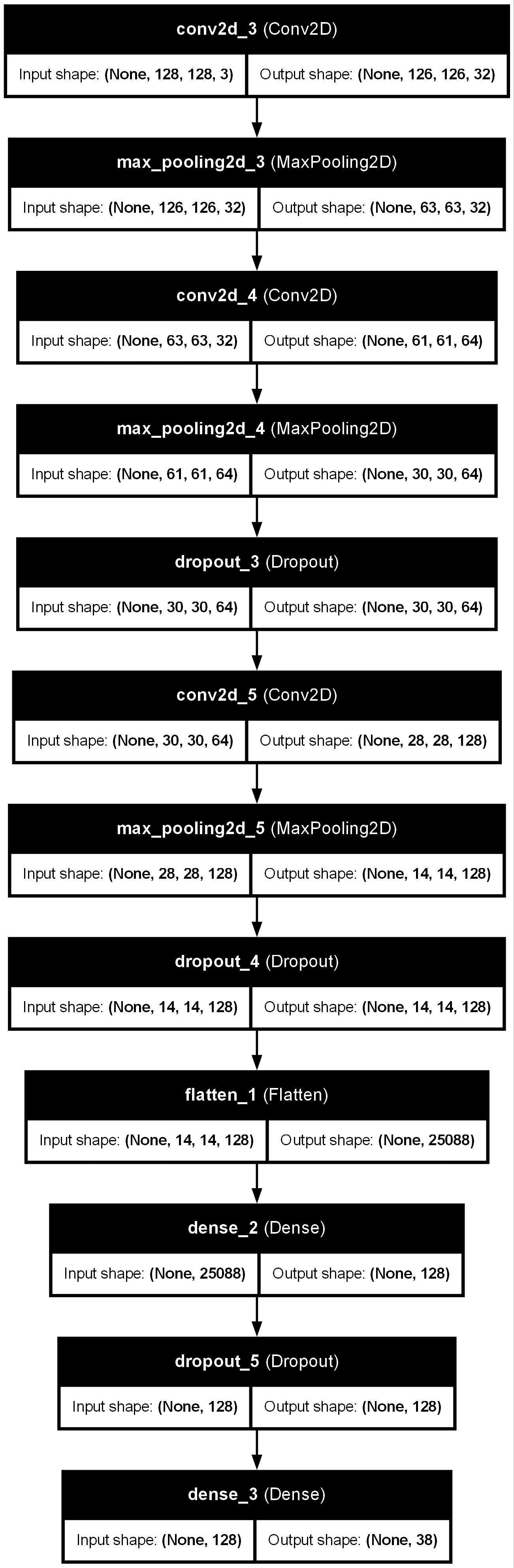
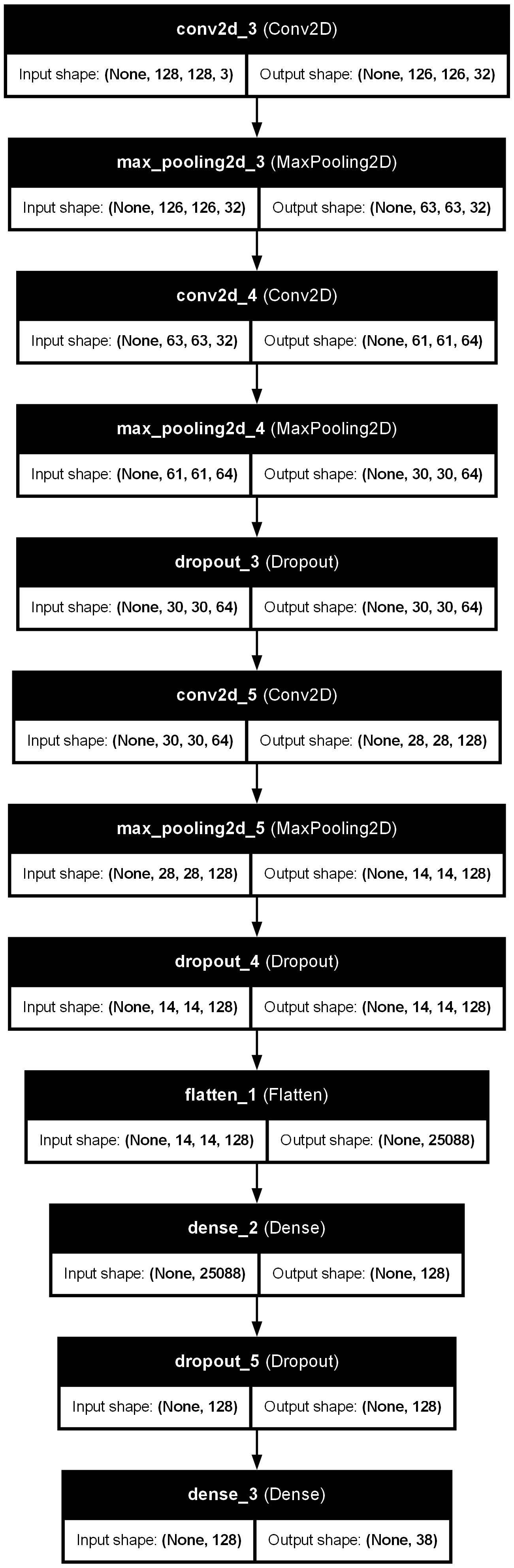
* Found 33 images with test class
* The size of the image is 128x128 and 3 channel.

**Data processing:**

The images are processed using the **ImageDataGenerator** from tensorflow to rescale the image, rotate, zoom and shear image.

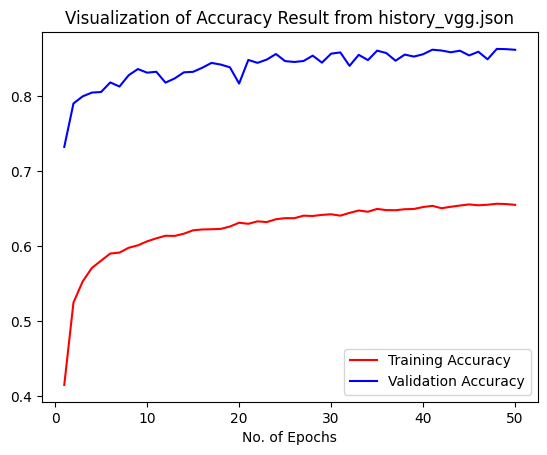
Once the images are fetched from the directory they are split into batches and specify the class mode. In our case the class mode will be categorical and batch size=32.

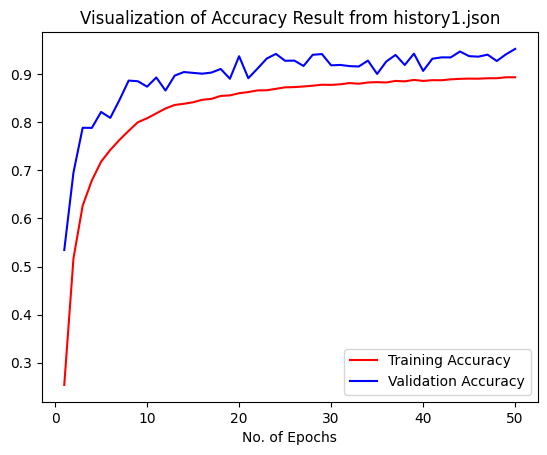
**Model building:**

1. Basic CNN Architecture
2. CNN Model with Increased Epochs
3. VGG model (Pre trained Model)
4. **Basic CNN Architecture Model:**
   1. The below is the CNN architecture for the first model
   2. The Epochs used for the training is 30 and training accuracy is 0.9316 and Validation accuracy is 0.9336
   3. The training history is saved as **‘histroy.json’** and model saved as **‘best\_model.h5’** and **‘Plant Disease CNN.keras’**
5. **CNN Model with Increased Epochs:**
   1. The below is the CNN architecture for the first model 
   2. The Epochs used for the training is 50 and training accuracy is 0.9430 and Validation accuracy is 0.9522
   3. The training history is saved as **‘histroy1.json’** and model saved as **‘best\_model1.h5’** and **‘Plant Disease CNN2.keras’**
6. **Pre Trained model (VGG):**
   1. The architecture for the VGG16 model is below: 
   2. The Epochs used for the training is 50 and training accuracy is 0.8400 and Validation accuracy is 0.8615
   3. The training history is saved as **‘histroy\_vgg.json’** and model saved as **‘best\_model\_vgg.h5’** and **‘Plant Disease CNN\_VGG.keras’**

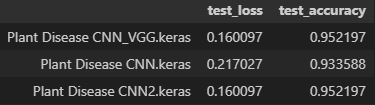
**Model Selection based on accuracy comparisons:**

1. **Train vs Validation Accuracy for the models:**
   1. For the VGG model:

****

* 1. For the Increased Epochs CNN model:****

1. Test metrics for the models:

****

Based on the above comparisons we can see the CNN architecture model performance is better in all train, valid and test data. Whereas the VGG performance is poor in train and valid data. Hence we will be using the "**best\_model1.h5**" model for the Streamlit application as well

**Streamlit application:**

1. **Libraries Used:**
   1. Streamlit
   2. Numpy
   3. Pandas
   4. Pillow
   5. Tensorflow
2. **Model Selection:**

The model is loaded to the application using the **tensorflow.keras.models.load\_model** as variable named **model.**

1. The class names are stored in variable **class\_names** which is extracted from the validation data generator classes.
2. Using **load\_and\_process\_image** function, the image uploaded is opened, resized and rescaled in expected format
3. Using **file\_uploader** function of streamlit, we are accepting the images of jpg, jpeg and png formats.
4. Once the file is uploaded, the model will be predicting the image and provides the prediction next to the uploaded image.

**Conclusion:**

The CNN model for the Plant disease detection from leaf images is developed successfully and we are getting an accuracy of 95.22 and implemented it using a Streamlit application for users.