SMARTBRIDGE
Let's Bridge the Gap
a Veranda Enterprise

Smart Internz

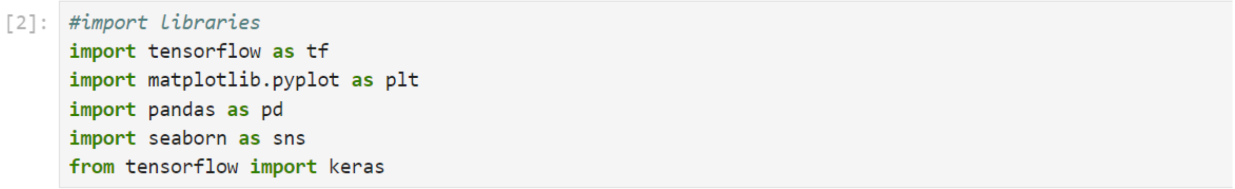
Model Development Phase Template

Date	7 July 2024
Team ID	SWTID1720076593
Project Title	Visual Diagnostics: Detecting Tomato Plant Diseases through Leaf Image Analysis
Maximum Marks	10 Marks

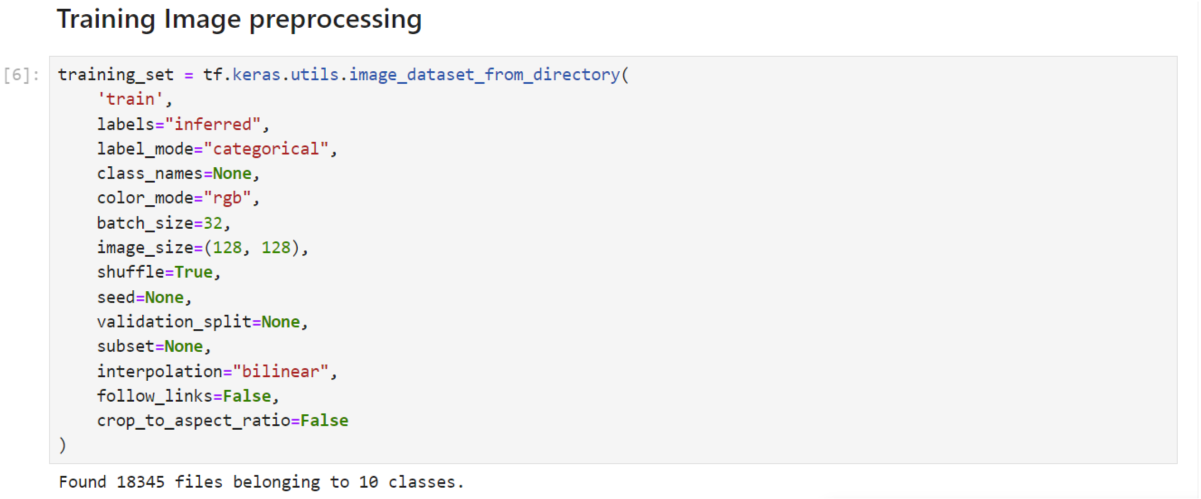
Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):



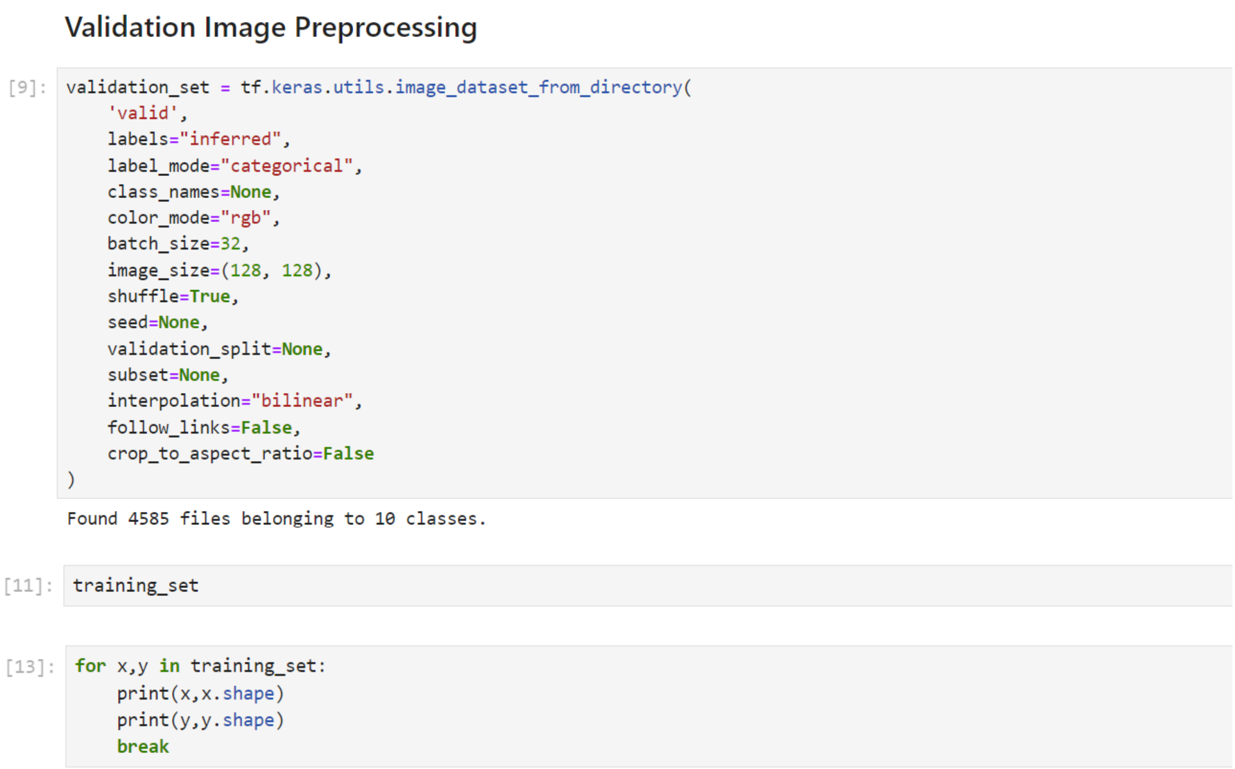
```
[2]: #import libraries
     import tensorflow as tf
     import matplotlib.pyplot as plt
     import pandas as pd
     import seaborn as sns
     from tensorflow import keras
```



Training Image preprocessing

```
[6]: training_set = tf.keras.utils.image_dataset_from_directory(
         'train',
         labels="inferred",
         label_mode="categorical",
         class_names=None,
         color_mode="rgb",
         batch_size=32,
         image_size=(128, 128),
         shuffle=True,
         seed=None,
         validation_split=None,
         subset=None,
         interpolation="bilinear",
         follow_links=False,
         crop_to_aspect_ratio=False
     )
```

```
Found 18345 files belonging to 10 classes.
```



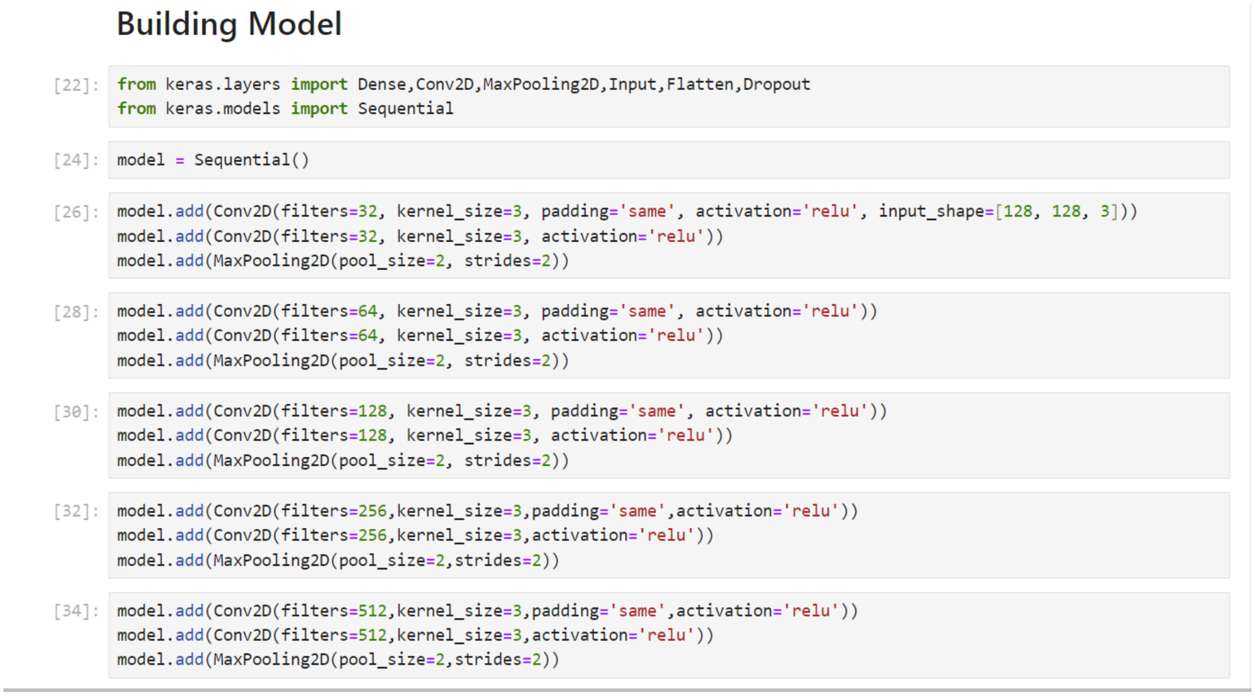
Validation Image Preprocessing

```
[9]: validation_set = tf.keras.utils.image_dataset_from_directory(
         'valid',
         labels="inferred",
         label_mode="categorical",
         class_names=None,
         color_mode="rgb",
         batch_size=32,
         image_size=(128, 128),
         shuffle=True,
         seed=None,
         validation_split=None,
         subset=None,
         interpolation="bilinear",
         follow_links=False,
         crop_to_aspect_ratio=False
     )
```

```
Found 4585 files belonging to 10 classes.
```

```
[11]: training_set
```

```
[13]: for x,y in training_set:
          print(x,x.shape)
          print(y,y.shape)
          break
```



Building Model

```
[22]: from keras.layers import Dense,Conv2D,MaxPooling2D,Input,Flatten,Dropout
      from keras.models import Sequential
```

```
[24]: model = Sequential()
```

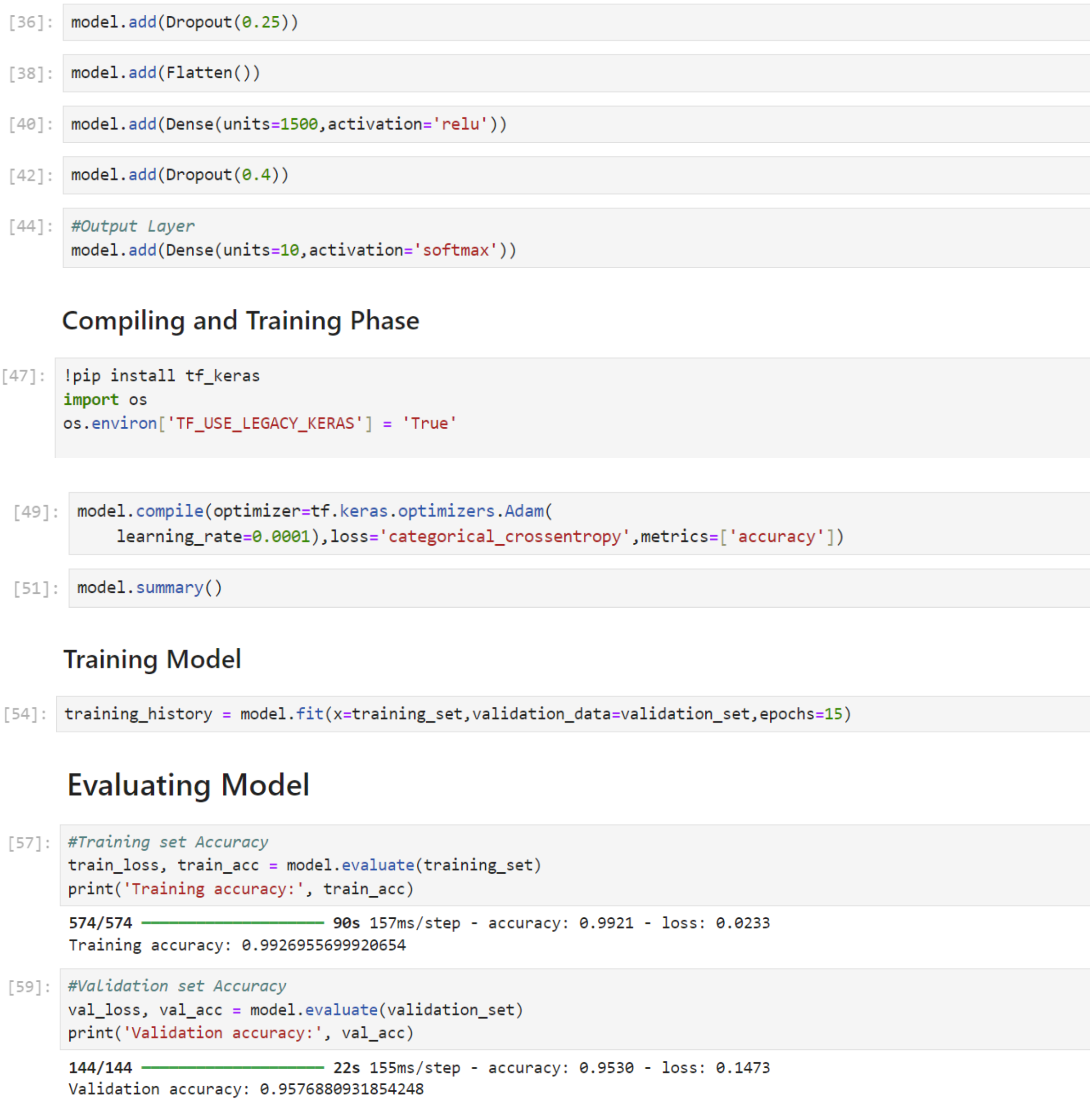
```
[26]: model.add(Conv2D(filters=32, kernel_size=3, padding='same', activation='relu', input_shape=[128, 128, 3]))
      model.add(Conv2D(filters=32, kernel_size=3, activation='relu'))
      model.add(MaxPooling2D(pool_size=2, strides=2))
```

```
[28]: model.add(Conv2D(filters=64, kernel_size=3, padding='same', activation='relu'))
      model.add(Conv2D(filters=64, kernel_size=3, activation='relu'))
      model.add(MaxPooling2D(pool_size=2, strides=2))
```

```
[30]: model.add(Conv2D(filters=128, kernel_size=3, padding='same', activation='relu'))
      model.add(Conv2D(filters=128, kernel_size=3, activation='relu'))
      model.add(MaxPooling2D(pool_size=2, strides=2))
```

```
[32]: model.add(Conv2D(filters=256,kernel_size=3,padding='same',activation='relu'))
      model.add(Conv2D(filters=256,kernel_size=3,activation='relu'))
      model.add(MaxPooling2D(pool_size=2,strides=2))
```

```
[34]: model.add(Conv2D(filters=512,kernel_size=3,padding='same',activation='relu'))
      model.add(Conv2D(filters=512,kernel_size=3,activation='relu'))
      model.add(MaxPooling2D(pool_size=2,strides=2))
```



[36]:
```
model.add(Dropout(0.25))
```

[38]:
```
model.add(Flatten())
```

[40]:
```
model.add(Dense(units=1500,activation='relu'))
```

[42]:
```
model.add(Dropout(0.4))
```

[44]:
```
#Output Layer
model.add(Dense(units=10,activation='softmax'))
```

Compiling and Training Phase

[47]:
```
!pip install tf_keras
import os
os.environ['TF_USE_LEGACY_KERAS'] = 'True'
```

[49]:
```
model.compile(optimizer=tf.keras.optimizers.Adam(
    learning_rate=0.0001),loss='categorical_crossentropy',metrics=['accuracy'])
```

[51]:
```
model.summary()
```

Training Model

[54]:
```
training_history = model.fit(x=training_set,validation_data=validation_set,epochs=15)
```

Evaluating Model

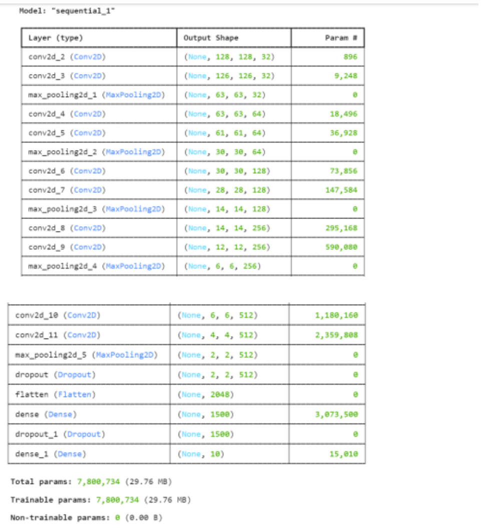
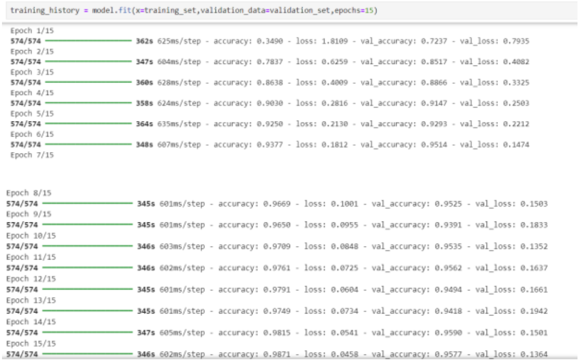
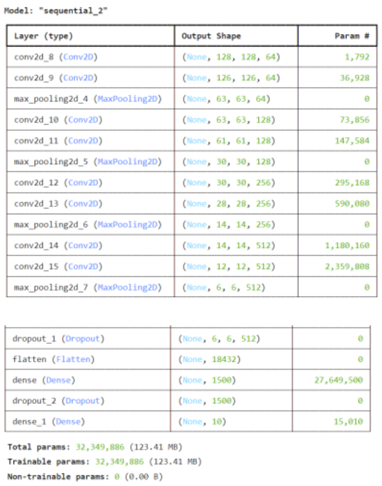
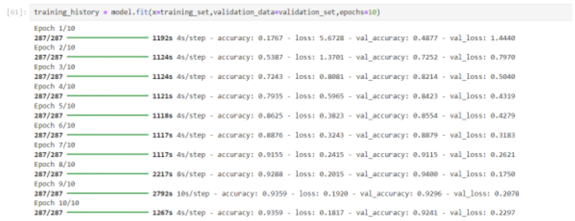
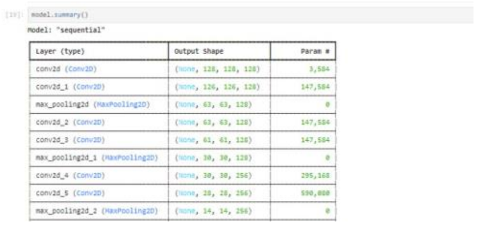
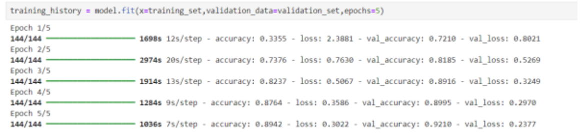
[57]:
```
#Training set Accuracy
train_loss, train_acc = model.evaluate(training_set)
print('Training accuracy:', train_acc)
```

```
574/574 ━━━━━━━━━━━━━━━━━━━━ 90s 157ms/step - accuracy: 0.9921 - loss: 0.0233
Training accuracy: 0.9926955699920654
```

[59]:
```
#Validation set Accuracy
val_loss, val_acc = model.evaluate(validation_set)
print('Validation accuracy:', val_acc)
```

```
144/144 ━━━━━━━━━━━━━━━━━━━━ 22s 155ms/step - accuracy: 0.9530 - loss: 0.1473
Validation accuracy: 0.9576880931854248
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
Model 1		
Model 2		
Model 3		

Model 1 – Summary:

```
Model: "sequential_1"

| Layer (type)                   | Output Shape          |     Param # |
|--------------------------------|-----------------------|-------------|
| conv2d_2 (Conv2D)              | (None, 128, 128, 32)  |         896 |
| conv2d_3 (Conv2D)              | (None, 126, 126, 32)  |       9,248 |
| max_pooling2d_1 (MaxPooling2D) | (None, 63, 63, 32)    |           0 |
| conv2d_4 (Conv2D)              | (None, 63, 63, 64)    |      18,496 |
| conv2d_5 (Conv2D)              | (None, 61, 61, 64)    |      36,928 |
| max_pooling2d_2 (MaxPooling2D) | (None, 30, 30, 64)    |           0 |
| conv2d_6 (Conv2D)              | (None, 30, 30, 128)   |      73,856 |
| conv2d_7 (Conv2D)              | (None, 28, 28, 128)   |     147,584 |
| max_pooling2d_3 (MaxPooling2D) | (None, 14, 14, 128)   |           0 |
| conv2d_8 (Conv2D)              | (None, 14, 14, 256)   |     295,168 |
| conv2d_9 (Conv2D)              | (None, 12, 12, 256)   |     590,080 |
| max_pooling2d_4 (MaxPooling2D) | (None, 6, 6, 256)     |           0 |

| conv2d_10 (Conv2D)             | (None, 6, 6, 512)     |   1,180,160 |
| conv2d_11 (Conv2D)             | (None, 4, 4, 512)     |   2,359,808 |
| max_pooling2d_5 (MaxPooling2D) | (None, 2, 2, 512)     |           0 |
| dropout (Dropout)              | (None, 2, 2, 512)     |           0 |
| flatten (Flatten)              | (None, 2048)          |           0 |
| dense (Dense)                  | (None, 1500)          |   3,073,500 |
| dropout_1 (Dropout)            | (None, 1500)          |           0 |
| dense_1 (Dense)                | (None, 10)            |      15,010 |

Total params: 7,800,734 (29.76 MB)
Trainable params: 7,800,734 (29.76 MB)
Non-trainable params: 0 (0.00 B)
```

Model 1 – Training and Validation Performance Metrics:

```
training_history = model.fit(x=training_set,validation_data=validation_set,epochs=15)
Epoch 1/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 362s 625ms/step - accuracy: 0.3490 - loss: 1.8109 - val_accuracy: 0.7237 - val_loss: 0.7935
Epoch 2/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 347s 604ms/step - accuracy: 0.7837 - loss: 0.6259 - val_accuracy: 0.8517 - val_loss: 0.4082
Epoch 3/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 360s 628ms/step - accuracy: 0.8638 - loss: 0.4009 - val_accuracy: 0.8866 - val_loss: 0.3325
Epoch 4/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 358s 624ms/step - accuracy: 0.9030 - loss: 0.2816 - val_accuracy: 0.9147 - val_loss: 0.2503
Epoch 5/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 364s 635ms/step - accuracy: 0.9250 - loss: 0.2130 - val_accuracy: 0.9293 - val_loss: 0.2212
Epoch 6/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 348s 607ms/step - accuracy: 0.9377 - loss: 0.1812 - val_accuracy: 0.9514 - val_loss: 0.1474
Epoch 7/15

Epoch 8/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 345s 601ms/step - accuracy: 0.9669 - loss: 0.1001 - val_accuracy: 0.9525 - val_loss: 0.1503
Epoch 9/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 345s 601ms/step - accuracy: 0.9650 - loss: 0.0955 - val_accuracy: 0.9391 - val_loss: 0.1833
Epoch 10/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 346s 603ms/step - accuracy: 0.9709 - loss: 0.0848 - val_accuracy: 0.9535 - val_loss: 0.1352
Epoch 11/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 346s 602ms/step - accuracy: 0.9761 - loss: 0.0725 - val_accuracy: 0.9562 - val_loss: 0.1637
Epoch 12/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 345s 601ms/step - accuracy: 0.9791 - loss: 0.0604 - val_accuracy: 0.9494 - val_loss: 0.1661
Epoch 13/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 345s 601ms/step - accuracy: 0.9749 - loss: 0.0734 - val_accuracy: 0.9418 - val_loss: 0.1942
Epoch 14/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 347s 605ms/step - accuracy: 0.9815 - loss: 0.0541 - val_accuracy: 0.9590 - val_loss: 0.1501
Epoch 15/15
574/574 ━━━━━━━━━━━━━━━━━━━━ 346s 602ms/step - accuracy: 0.9871 - loss: 0.0458 - val_accuracy: 0.9577 - val_loss: 0.1364
```

Model 2 – Summary:

```
Model: "sequential_2"

| Layer (type)                   | Output Shape          |     Param # |
|--------------------------------|-----------------------|-------------|
| conv2d_8 (Conv2D)              | (None, 128, 128, 64)  |       1,792 |
| conv2d_9 (Conv2D)              | (None, 126, 126, 64)  |      36,928 |
| max_pooling2d_4 (MaxPooling2D) | (None, 63, 63, 64)    |           0 |
| conv2d_10 (Conv2D)             | (None, 63, 63, 128)   |      73,856 |
| conv2d_11 (Conv2D)             | (None, 61, 61, 128)   |     147,584 |
| max_pooling2d_5 (MaxPooling2D) | (None, 30, 30, 128)   |           0 |
| conv2d_12 (Conv2D)             | (None, 30, 30, 256)   |     295,168 |
| conv2d_13 (Conv2D)             | (None, 28, 28, 256)   |     590,080 |
| max_pooling2d_6 (MaxPooling2D) | (None, 14, 14, 256)   |           0 |
| conv2d_14 (Conv2D)             | (None, 14, 14, 512)   |   1,180,160 |
| conv2d_15 (Conv2D)             | (None, 12, 12, 512)   |   2,359,808 |
| max_pooling2d_7 (MaxPooling2D) | (None, 6, 6, 512)     |           0 |

| dropout_1 (Dropout)            | (None, 6, 6, 512)     |           0 |
| flatten (Flatten)              | (None, 18432)         |           0 |
| dense (Dense)                  | (None, 1500)          |  27,649,500 |
| dropout_2 (Dropout)            | (None, 1500)          |           0 |
| dense_1 (Dense)                | (None, 10)            |      15,010 |

Total params: 32,349,886 (123.41 MB)
Trainable params: 32,349,886 (123.41 MB)
Non-trainable params: 0 (0.00 B)
```

Model 2 – Training and Validation Performance Metrics:

```
[61]: training_history = model.fit(x=training_set,validation_data=validation_set,epochs=10)
Epoch 1/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1192s 4s/step - accuracy: 0.1767 - loss: 5.6728 - val_accuracy: 0.4877 - val_loss: 1.4440
Epoch 2/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1124s 4s/step - accuracy: 0.5387 - loss: 1.3701 - val_accuracy: 0.7252 - val_loss: 0.7970
Epoch 3/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1124s 4s/step - accuracy: 0.7243 - loss: 0.8081 - val_accuracy: 0.8214 - val_loss: 0.5040
Epoch 4/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1121s 4s/step - accuracy: 0.7935 - loss: 0.5965 - val_accuracy: 0.8423 - val_loss: 0.4319
Epoch 5/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1118s 4s/step - accuracy: 0.8625 - loss: 0.3823 - val_accuracy: 0.8554 - val_loss: 0.4279
Epoch 6/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1117s 4s/step - accuracy: 0.8876 - loss: 0.3243 - val_accuracy: 0.8879 - val_loss: 0.3183
Epoch 7/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1117s 4s/step - accuracy: 0.9155 - loss: 0.2415 - val_accuracy: 0.9115 - val_loss: 0.2621
Epoch 8/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 2217s 8s/step - accuracy: 0.9288 - loss: 0.2015 - val_accuracy: 0.9400 - val_loss: 0.1750
Epoch 9/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 2792s 10s/step - accuracy: 0.9359 - loss: 0.1920 - val_accuracy: 0.9296 - val_loss: 0.2078
Epoch 10/10
287/287 ━━━━━━━━━━━━━━━━━━━━ 1267s 4s/step - accuracy: 0.9359 - loss: 0.1817 - val_accuracy: 0.9241 - val_loss: 0.2297
```

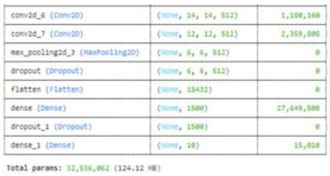
Model 3 – Summary:

```
[19]: model.summary()
Model: "sequential"

| Layer (type)                   | Output Shape          |     Param # |
|--------------------------------|-----------------------|-------------|
| conv2d (Conv2D)                | (None, 128, 128, 128) |       3,584 |
| conv2d_1 (Conv2D)              | (None, 126, 126, 128) |     147,584 |
| max_pooling2d (MaxPooling2D)   | (None, 63, 63, 128)   |           0 |
| conv2d_2 (Conv2D)              | (None, 63, 63, 128)   |     147,584 |
| conv2d_3 (Conv2D)              | (None, 61, 61, 128)   |     147,584 |
| max_pooling2d_1 (MaxPooling2D) | (None, 30, 30, 128)   |           0 |
| conv2d_4 (Conv2D)              | (None, 30, 30, 256)   |     295,168 |
| conv2d_5 (Conv2D)              | (None, 28, 28, 256)   |     590,080 |
| max_pooling2d_2 (MaxPooling2D) | (None, 14, 14, 256)   |           0 |
```

Model 3 – Training and Validation Performance Metrics:

```
training_history = model.fit(x=training_set,validation_data=validation_set,epochs=5)
Epoch 1/5
144/144 ━━━━━━━━━━━━━━━━━━━━ 1698s 12s/step - accuracy: 0.3355 - loss: 2.3881 - val_accuracy: 0.7210 - val_loss: 0.8021
Epoch 2/5
144/144 ━━━━━━━━━━━━━━━━━━━━ 2974s 20s/step - accuracy: 0.7376 - loss: 0.7630 - val_accuracy: 0.8185 - val_loss: 0.5269
Epoch 3/5
144/144 ━━━━━━━━━━━━━━━━━━━━ 1914s 13s/step - accuracy: 0.8237 - loss: 0.5067 - val_accuracy: 0.8916 - val_loss: 0.3249
Epoch 4/5
144/144 ━━━━━━━━━━━━━━━━━━━━ 1284s 9s/step - accuracy: 0.8764 - loss: 0.3586 - val_accuracy: 0.8995 - val_loss: 0.2970
Epoch 5/5
144/144 ━━━━━━━━━━━━━━━━━━━━ 1036s 7s/step - accuracy: 0.8942 - loss: 0.3022 - val_accuracy: 0.9210 - val_loss: 0.2377
```

	 conv2d_6 (Conv2D) (None, 14, 14, 512) 1,180,160 conv2d_7 (Conv2D) (None, 12, 12, 512) 2,359,808 max_pooling2d_3 (MaxPooling2D) (None, 6, 6, 512) 0 dropout (Dropout) (None, 6, 6, 512) 0 flatten (Flatten) (None, 18432) 0 dense (Dense) (None, 1500) 27,649,500 dropout_1 (Dropout) (None, 1500) 0 dense_1 (Dense) (None, 10) 15,010 Total params: 32,536,062 (124.12 MB)	