Model Development Phase Template

Date	22 April 2024
Team ID	**738194**
Project Title	RIPE-SENSE: MANGO QUALITY GRADING WITH IMAGE ANALYSIS AND DEEP LEARNING.
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 screenshot.

Initial Model Training Code (5 marks):

VGG16:

```
17]: from tensorflow.keras.applications.vgg16 import VGG16
     from tensorflow.keras.layers import Dense,Flatten
     from tensorflow.keras.models import Model

29]: vgg = VGG16(include_top=False,input_shape=(128,128,3))
     vgg.trainalbe = False

30]: vgg.summary()

     Model: "vgg16"
     _________________________________________________________________
```

```
[37]: vgg16.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

[38]: history = vgg16.fit(train_generator,validation_data=validation_generator,epochs=100)

Epoch 92/100
18/18 [==============================] - 32s 2s/step - loss: 0.0655 - accuracy: 0.9944 - val_loss: 0.6607 - val_accuracy: 0.7417
Epoch 93/100
18/18 [==============================] - 33s 2s/step - loss: 0.0677 - accuracy: 0.9889 - val_loss: 0.5640 - val_accuracy: 0.7875
Epoch 94/100
18/18 [==============================] - 31s 2s/step - loss: 0.0857 - accuracy: 0.9722 - val_loss: 0.7011 - val_accuracy: 0.7583
Epoch 95/100
18/18 [==============================] - 30s 2s/step - loss: 0.0747 - accuracy: 0.9806 - val_loss: 0.5226 - val_accuracy: 0.7792
Epoch 96/100
18/18 [==============================] - 31s 2s/step - loss: 0.0722 - accuracy: 0.9889 - val_loss: 0.6842 - val_accuracy: 0.7458
Epoch 97/100
18/18 [==============================] - 31s 2s/step - loss: 0.0618 - accuracy: 0.9972 - val_loss: 0.6367 - val_accuracy: 0.7500
Epoch 98/100
18/18 [==============================] - 30s 2s/step - loss: 0.0622 - accuracy: 0.9944 - val_loss: 0.6284 - val_accuracy: 0.7542
Epoch 99/100
18/18 [==============================] - 31s 2s/step - loss: 0.0659 - accuracy: 0.9944 - val_loss: 0.6484 - val_accuracy: 0.7542
Epoch 100/100
18/18 [==============================] - 32s 2s/step - loss: 0.0647 - accuracy: 0.9917 - val_loss: 0.5774 - val_accuracy: 0.7708

[47]: #testing accuracy on training dataset
      loss, accuracy = vgg16.evaluate(test_generator)
      print("Test Accuracy : {:.2f}%".format(accuracy*100))

      30/30 [==============================] - 27s 880ms/step - loss: 0.3550 - accuracy: 0.8767
      Test Accuracy : 87.67%

[46]: #TESTING Accuracy on valisation dataset
      loss, accuracy = vgg16.evaluate(validation_generator)
      print("Validation Accuracy : {:.2f}%".format(accuracy*100))

      12/12 [==============================] - 11s 889ms/step - loss: 0.5621 - accuracy: 0.7833
      Validation Accuracy : 78.33%
```

CNN (Sequential) :

```
#importing the model building libraries
from tensorflow.keras import layers
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.initializers import glorot_uniform, glorot_normal, he_uniform
from sklearn.metrics import classification_report, confusion_matrix, precision_recall_curve
import seaborn as sns
from keras.models import Model
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import CategoricalCrossentropy
from tensorflow.keras.metrics import Accuracy
```

```
model = Sequential()
model.add(Convolution2D((64),(3,3),kernel_initializer=glorot_uniform(seed=10),input_shape = (128,128,3),activation="relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(units=64,kernel_initializer=he_uniform(seed=10),activation='relu')) #first Hidden layer
model.add(Dense(units=64,kernel_initializer=he_uniform(seed=10),activation='relu')) #second hidden layer
model.add(Dense(units=3,kernel_initializer=he_uniform(seed=10),activation='softmax'))
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:99: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(
```

```
model.summary()
```

```
Model: "sequential"
```

```
model.compile(optimizer = 'rmsprop',loss='categorical_crossentropy',metrics=["accuracy"])
```

```
history = model.fit(train_generator,epochs =100, validation_data = validation_generator)
```

```
Epoch 92/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 540ms/step - accuracy: 0.9305 - loss: 0.1885 - val_accuracy: 0.6917 - val_loss: 1.1080
Epoch 93/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 17s 598ms/step - accuracy: 0.9341 - loss: 0.1517 - val_accuracy: 0.5083 - val_loss: 3.2152
Epoch 94/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 525ms/step - accuracy: 0.7848 - loss: 0.8802 - val_accuracy: 0.7417 - val_loss: 0.9215
Epoch 95/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 532ms/step - accuracy: 0.9472 - loss: 0.1080 - val_accuracy: 0.6083 - val_loss: 1.4439
Epoch 96/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 524ms/step - accuracy: 0.8901 - loss: 0.2393 - val_accuracy: 0.7000 - val_loss: 1.2387
Epoch 97/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 15s 514ms/step - accuracy: 0.9232 - loss: 0.2413 - val_accuracy: 0.6667 - val_loss: 1.3673
Epoch 98/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 15s 506ms/step - accuracy: 0.9594 - loss: 0.1133 - val_accuracy: 0.7167 - val_loss: 1.0998
Epoch 99/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 546ms/step - accuracy: 0.9425 - loss: 0.1839 - val_accuracy: 0.6750 - val_loss: 1.1386
Epoch 100/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 16s 539ms/step - accuracy: 0.8976 - loss: 0.2280 - val_accuracy: 0.7417 - val_loss: 1.0512
```

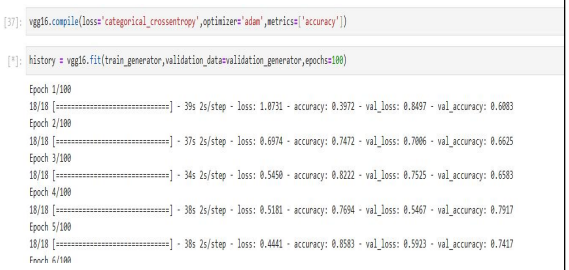
```
#TESTING Accuracy on valisation dataset
loss, accuracy = model.evaluate(validation_generator)
print("Validation Accuracy : {:.2f}%".format(accuracy*100))
```

```
6/6 ━━━━━━━━━━━━━━━━━━━━ 3s 446ms/step - accuracy: 0.7385 - loss: 0.8158
Validation Accuracy : 72.50%
```

```
#testing accuracy on training dataset
loss, accuracy = model.evaluate(test_generator)
print("Test Accuracy : {:.2f}%".format(accuracy*100))
```

```
30/30 ━━━━━━━━━━━━━━━━━━━━ 8s 270ms/step - accuracy: 0.9445 - loss: 0.2349
Test Accuracy : 93.33%
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
VGG16		
CNN (Sequential)