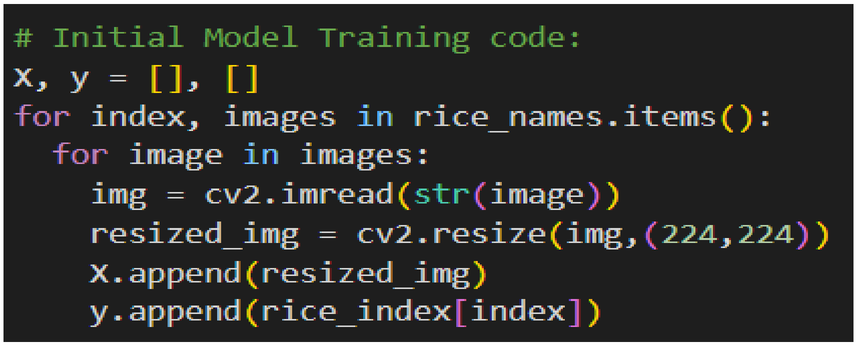
Model Development Phase Template

Date	08 July 2024
Team ID	SWTID1720201335
Project Title	Rice Type Classification Using Cnn
Maximum Marks	10 Marks

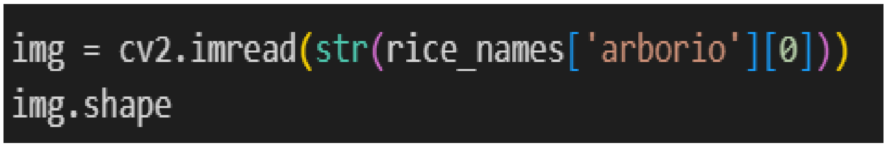
Initial Model Training Code, Model Validation and Evaluation Report

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.

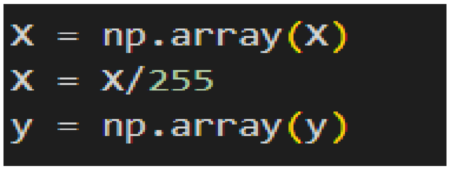
Initial Model Training Code (5 marks):



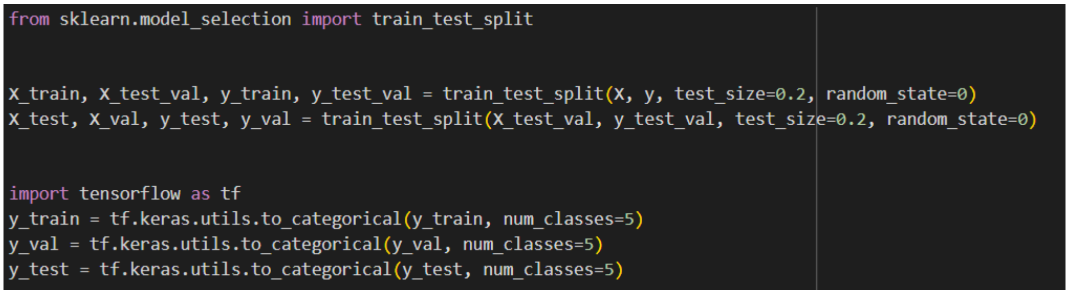
```
# Initial Model Training code:
X, y = [], []
for index, images in rice_names.items():
  for image in images:
    img = cv2.imread(str(image))
    resized_img = cv2.resize(img,(224,224))
    X.append(resized_img)
    y.append(rice_index[index])
```



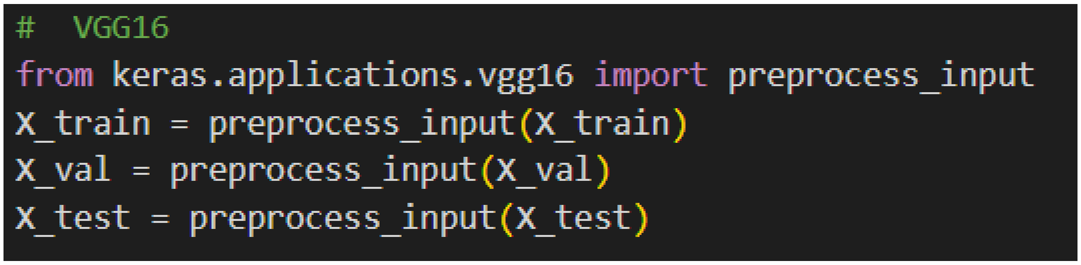
```
img = cv2.imread(str(rice_names['arborio'][0]))
img.shape
```



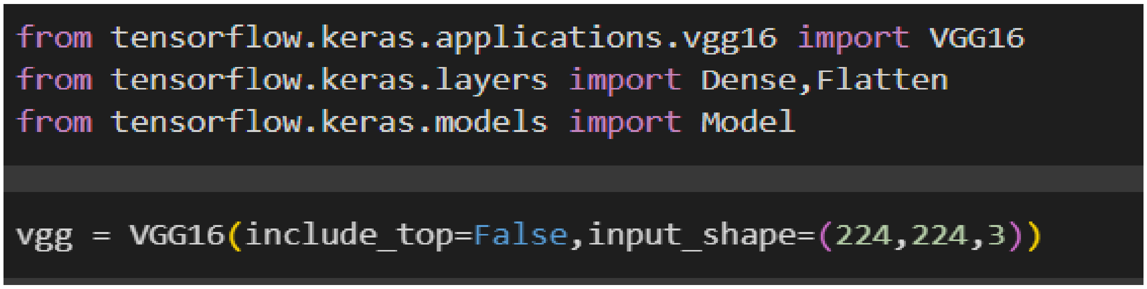
```
X = np.array(X)
X = X/255
y = np.array(y)
```



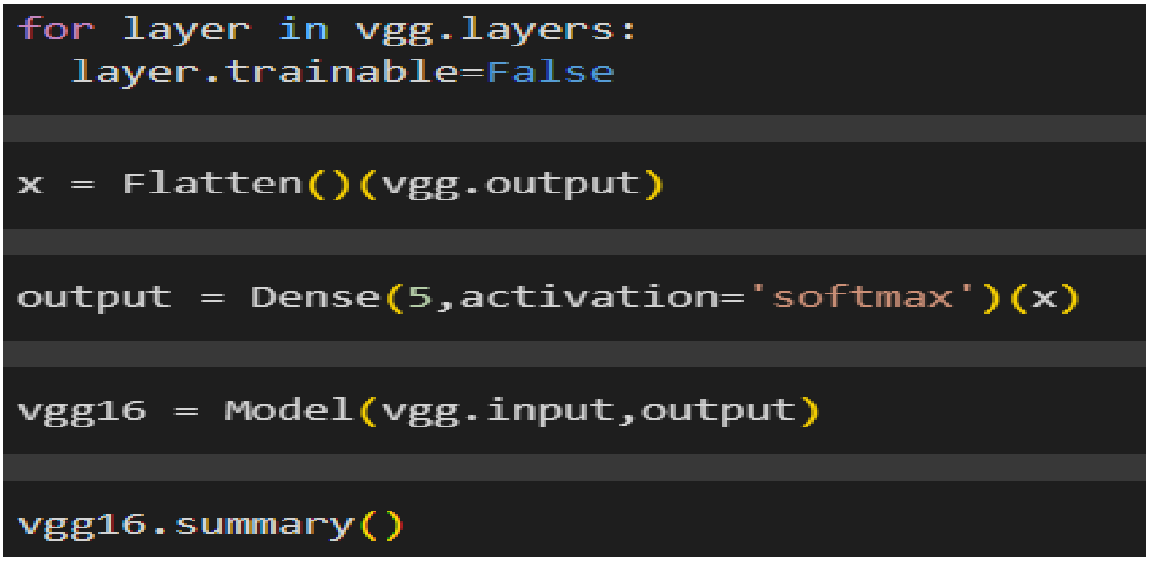
```
from sklearn.model_selection import train_test_split


X_train, X_test_val, y_train, y_test_val = train_test_split(X, y, test_size=0.2, random_state=0)
X_test, X_val, y_test, y_val = train_test_split(X_test_val, y_test_val, test_size=0.2, random_state=0)


import tensorflow as tf
y_train = tf.keras.utils.to_categorical(y_train, num_classes=5)
y_val = tf.keras.utils.to_categorical(y_val, num_classes=5)
y_test = tf.keras.utils.to_categorical(y_test, num_classes=5)
```



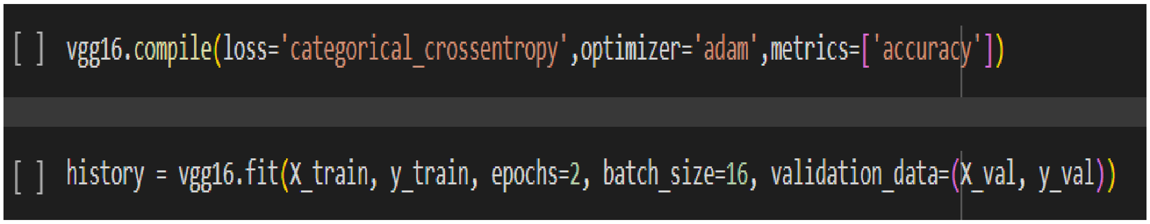
```
#  VGG16
from keras.applications.vgg16 import preprocess_input
X_train = preprocess_input(X_train)
X_val = preprocess_input(X_val)
X_test = preprocess_input(X_test)
```



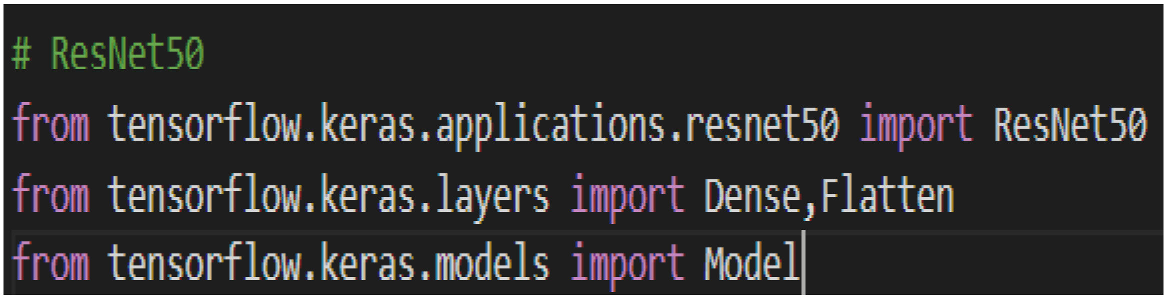
```
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.models import Model

vgg = VGG16(include_top=False,input_shape=(224,224,3))
```



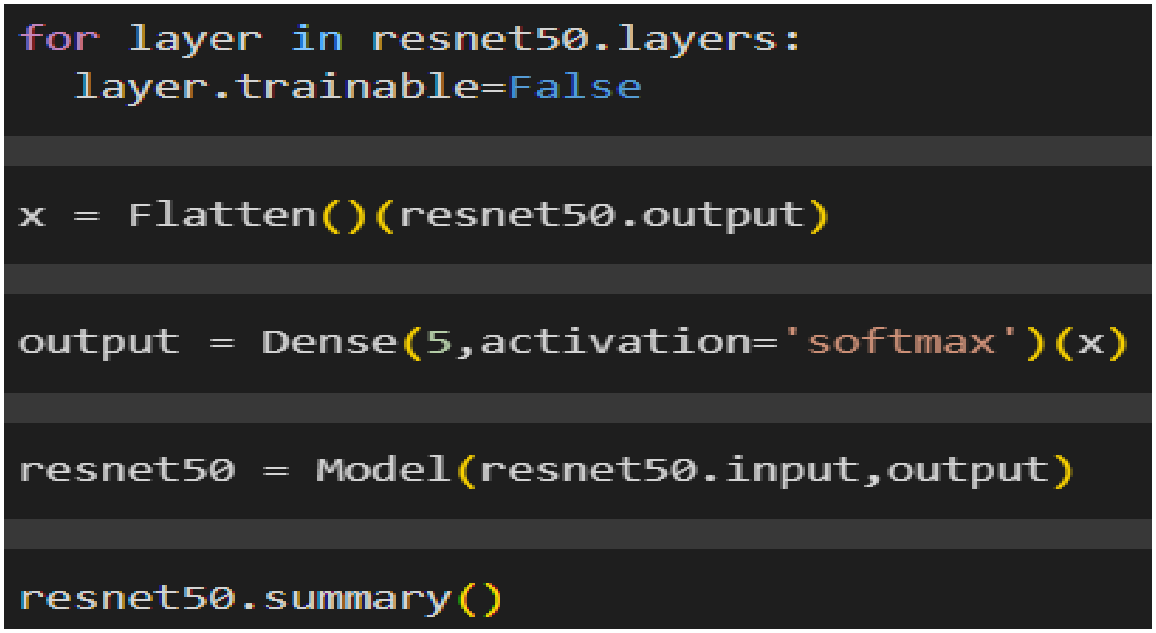
```
for layer in vgg.layers:
  layer.trainable=False

x = Flatten()(vgg.output)

output = Dense(5,activation='softmax')(x)

vgg16 = Model(vgg.input,output)

vgg16.summary()
```



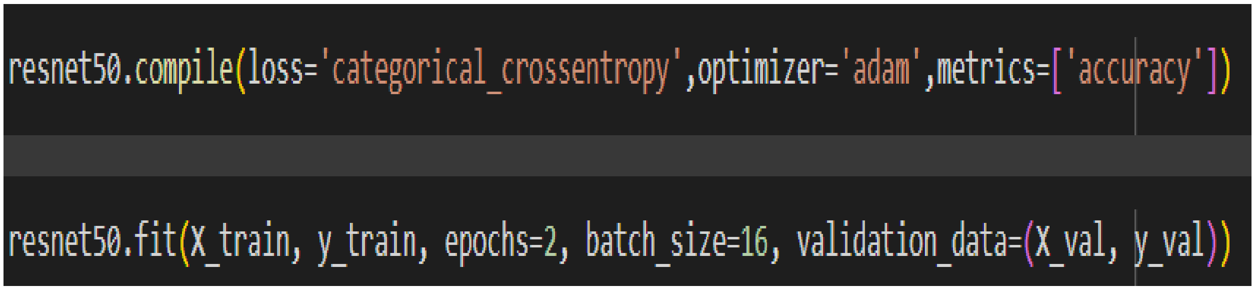
```
[ ]  vgg16.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

[ ]  history = vgg16.fit(X_train, y_train, epochs=2, batch_size=16, validation_data=(X_val, y_val))
```



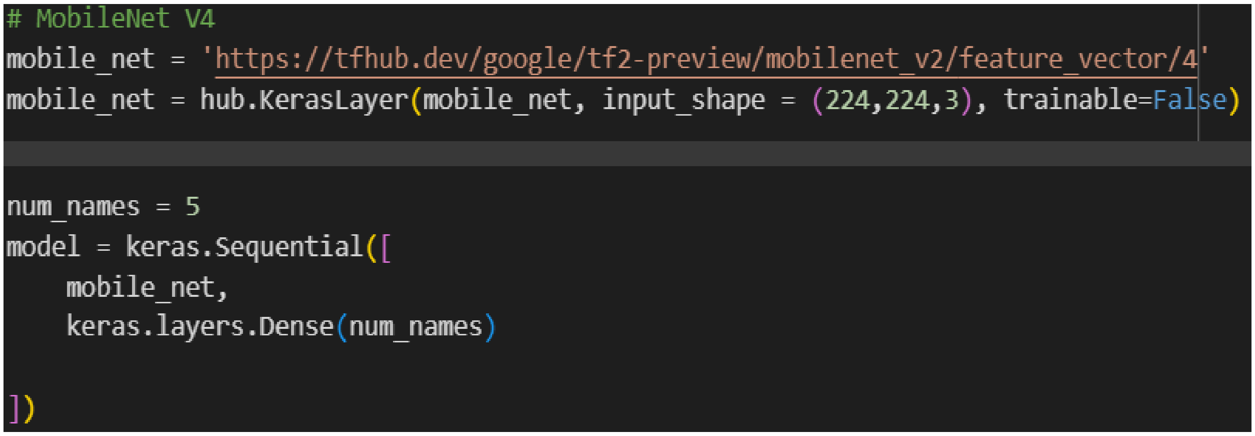
```
# ResNet50
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.models import Model
```



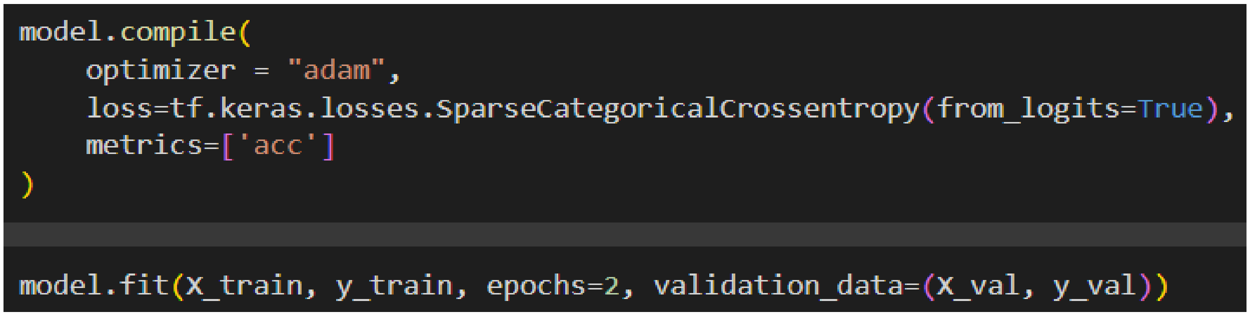
```
for layer in resnet50.layers:
  layer.trainable=False

x = Flatten()(resnet50.output)

output = Dense(5,activation='softmax')(x)

resnet50 = Model(resnet50.input,output)

resnet50.summary()
```



```
resnet50.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

resnet50.fit(X_train, y_train, epochs=2, batch_size=16, validation_data=(X_val, y_val))
```



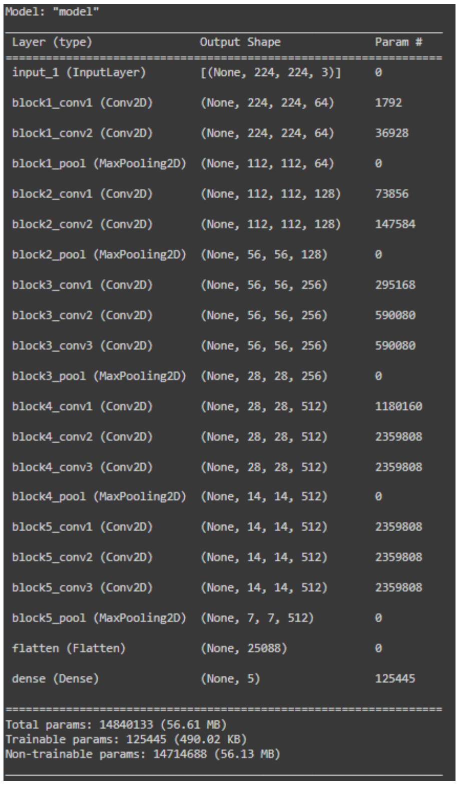
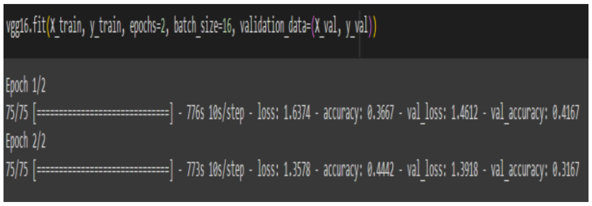
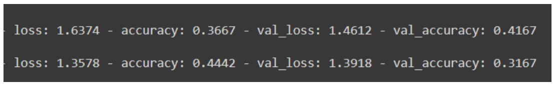
```
# MobileNet V4
mobile_net = 'https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4'
mobile_net = hub.KerasLayer(mobile_net, input_shape = (224,224,3), trainable=False)

num_names = 5
model = keras.Sequential([
    mobile_net,
    keras.layers.Dense(num_names)

])
```

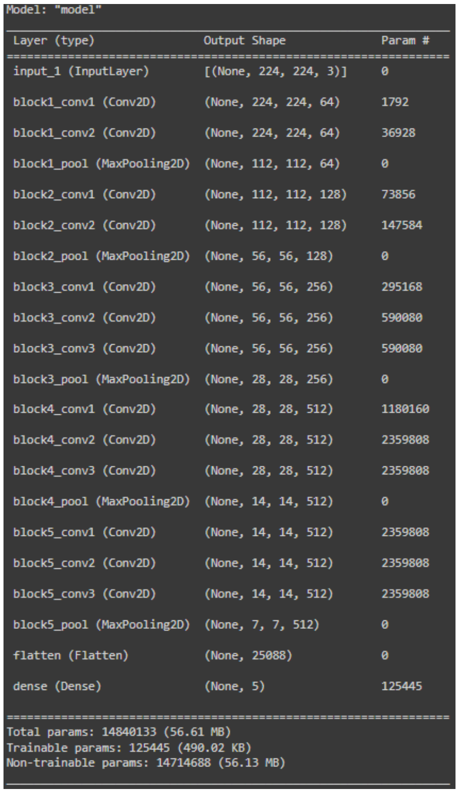
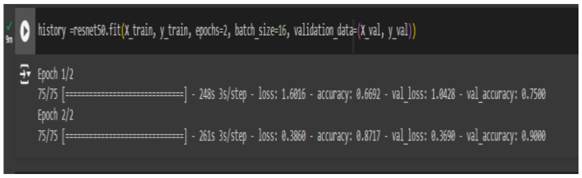
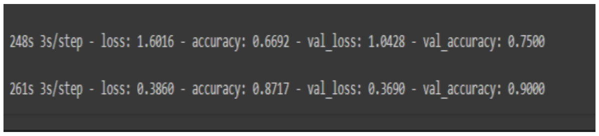
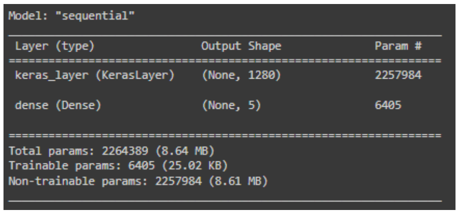
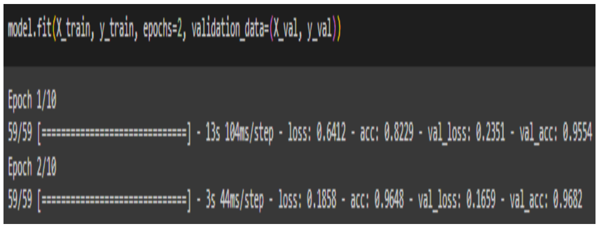
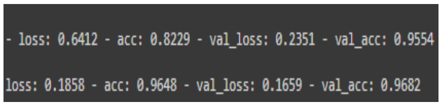


```
model.compile(
    optimizer = "adam",
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['acc']
)
```

```
model.fit(X_train, y_train, epochs=2, validation_data=(X_val, y_val))
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
VGG16	 Model: "model" Layer (type) Output Shape Param # input_1 (InputLayer) [(None, 224, 224, 3)] 0 block1_conv1 (Conv2D) (None, 224, 224, 64) 1792 block1_conv2 (Conv2D) (None, 224, 224, 64) 36928 block1_pool (MaxPooling2D) (None, 112, 112, 64) 0 block2_conv1 (Conv2D) (None, 112, 112, 128) 73856 block2_conv2 (Conv2D) (None, 112, 112, 128) 147584 block2_pool (MaxPooling2D) (None, 56, 56, 128) 0 block3_conv1 (Conv2D) (None, 56, 56, 256) 295168 block3_conv2 (Conv2D) (None, 56, 56, 256) 590080 block3_conv3 (Conv2D) (None, 56, 56, 256) 590080 block3_pool (MaxPooling2D) (None, 28, 28, 256) 0 block4_conv1 (Conv2D) (None, 28, 28, 512) 1180160 block4_conv2 (Conv2D) (None, 28, 28, 512) 2359808 block4_conv3 (Conv2D) (None, 28, 28, 512) 2359808 block4_pool (MaxPooling2D) (None, 14, 14, 512) 0 block5_conv1 (Conv2D) (None, 14, 14, 512) 2359808 block5_conv2 (Conv2D) (None, 14, 14, 512) 2359808 block5_conv3 (Conv2D) (None, 14, 14, 512) 2359808 block5_pool (MaxPooling2D) (None, 7, 7, 512) 0 flatten (Flatten) (None, 25088) 0 dense (Dense) (None, 5) 125445 Total params: 14840133 (56.61 MB) Trainable params: 125445 (490.02 KB) Non-trainable params: 14714688 (56.13 MB)	 vgg16.fit(X_train, y_train, epochs=2, batch_size=16, validation_data=(X_val, y_val)) Epoch 1/2 75/75 [==============================] - 776s 10s/step - loss: 1.6374 - accuracy: 0.3667 - val_loss: 1.4612 - val_accuracy: 0.4167 Epoch 2/2 75/75 [==============================] - 773s 10s/step - loss: 1.3578 - accuracy: 0.4442 - val_loss: 1.3918 - val_accuracy: 0.3167  loss: 1.6374 - accuracy: 0.3667 - val_loss: 1.4612 - val_accuracy: 0.4167 loss: 1.3578 - accuracy: 0.4442 - val_loss: 1.3918 - val_accuracy: 0.3167

ResNet50	 Model: "model" Layer (type) Output Shape Param # === input_1 (InputLayer) [(None, 224, 224, 3)] 0 block1_conv1 (Conv2D) (None, 224, 224, 64) 1792 block1_conv2 (Conv2D) (None, 224, 224, 64) 36928 block1_pool (MaxPooling2D) (None, 112, 112, 64) 0 block2_conv1 (Conv2D) (None, 112, 112, 128) 73856 block2_conv2 (Conv2D) (None, 112, 112, 128) 147584 block2_pool (MaxPooling2D) (None, 56, 56, 128) 0 block3_conv1 (Conv2D) (None, 56, 56, 256) 295168 block3_conv2 (Conv2D) (None, 56, 56, 256) 590080 block3_conv3 (Conv2D) (None, 56, 56, 256) 590080 block3_pool (MaxPooling2D) (None, 28, 28, 256) 0 block4_conv1 (Conv2D) (None, 28, 28, 512) 1180160 block4_conv2 (Conv2D) (None, 28, 28, 512) 2359808 block4_conv3 (Conv2D) (None, 28, 28, 512) 2359808 block4_pool (MaxPooling2D) (None, 14, 14, 512) 0 block5_conv1 (Conv2D) (None, 14, 14, 512) 2359808 block5_conv2 (Conv2D) (None, 14, 14, 512) 2359808 block5_conv3 (Conv2D) (None, 14, 14, 512) 2359808 block5_pool (MaxPooling2D) (None, 7, 7, 512) 0 flatten (Flatten) (None, 25088) 0 dense (Dense) (None, 5) 125445 === Total params: 14840133 (56.61 MB) Trainable params: 125445 (490.02 KB) Non-trainable params: 14714688 (56.13 MB)	 history =resnet50.fit(X_train, y_train, epochs=2, batch_size=16, validation_data=(X_val, y_val)) Epoch 1/2 75/75 [==============================] - 248s 3s/step - loss: 1.6016 - accuracy: 0.6692 - val_loss: 1.0428 - val_accuracy: 0.7500 Epoch 2/2 75/75 [==============================] - 261s 3s/step - loss: 0.3860 - accuracy: 0.8717 - val_loss: 0.3690 - val_accuracy: 0.9000  248s 3s/step - loss: 1.6016 - accuracy: 0.6692 - val_loss: 1.0428 - val_accuracy: 0.7500 261s 3s/step - loss: 0.3860 - accuracy: 0.8717 - val_loss: 0.3690 - val_accuracy: 0.9000
MobileNetV4	 Model: "sequential" Layer (type) Output Shape Param # === keras_layer (KerasLayer) (None, 1280) 2257984 dense (Dense) (None, 5) 6405 === Total params: 2264389 (8.64 MB) Trainable params: 6405 (25.02 KB) Non-trainable params: 2257984 (8.61 MB)	 model.fit(X_train, y_train, epochs=2, validation_data=(X_val, y_val)) Epoch 1/10 59/59 [==============================] - 13s 104ms/step - loss: 0.6412 - acc: 0.8229 - val_loss: 0.2351 - val_acc: 0.9554 Epoch 2/10 59/59 [==============================] - 3s 44ms/step - loss: 0.1858 - acc: 0.9648 - val_loss: 0.1659 - val_acc: 0.9682  - loss: 0.6412 - acc: 0.8229 - val_loss: 0.2351 - val_acc: 0.9554 loss: 0.1858 - acc: 0.9648 - val_loss: 0.1659 - val_acc: 0.9682