VGG16 FUNDUS lr=0.001

from keras.applications import VGG16, ResNet50, InceptionV3

from keras.models import Sequential

from keras.layers import Dense, Flatten

from keras.preprocessing.image import ImageDataGenerator

from keras.optimizers import Adam

# Set the dimensions of your input images

img\_width, img\_height = 100, 100

# Specify the pre-trained model you want to use

pretrained\_model = VGG16(weights='imagenet', include\_top=False, input\_shape=(img\_width, img\_height, 3))

#pretrained\_model = ResNet50(weights='imagenet', include\_top=False, input\_shape=(img\_width, img\_height, 3))

#pretrained\_model = InceptionV3(weights='imagenet', include\_top=False, input\_shape=(img\_width, img\_height, 3))

# Freeze the layers of the pre-trained model

for layer in pretrained\_model.layers:

    layer.trainable = False

# Build a new model and add the pre-trained model as a layer

model = Sequential()

model.add(pretrained\_model)

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dense(3, activation='softmax'))

# Compile the model

model.compile(optimizer=Adam(learning\_rate=0.001), loss='categorical\_crossentropy', metrics=['accuracy'],callbacks=[AccuracyCallback()])

# Specify the directories of your training and validation datasets

train\_dir = '/content/drive/MyDrive/Fundus\_Split/train'

val\_dir = '/content/drive/MyDrive/Fundus\_Split/test'

# Set up data generators to automatically preprocess and augment your images

train\_datagen = ImageDataGenerator(rescale=1.0/255.0, shear\_range=0.2, zoom\_range=0.2, horizontal\_flip=True)

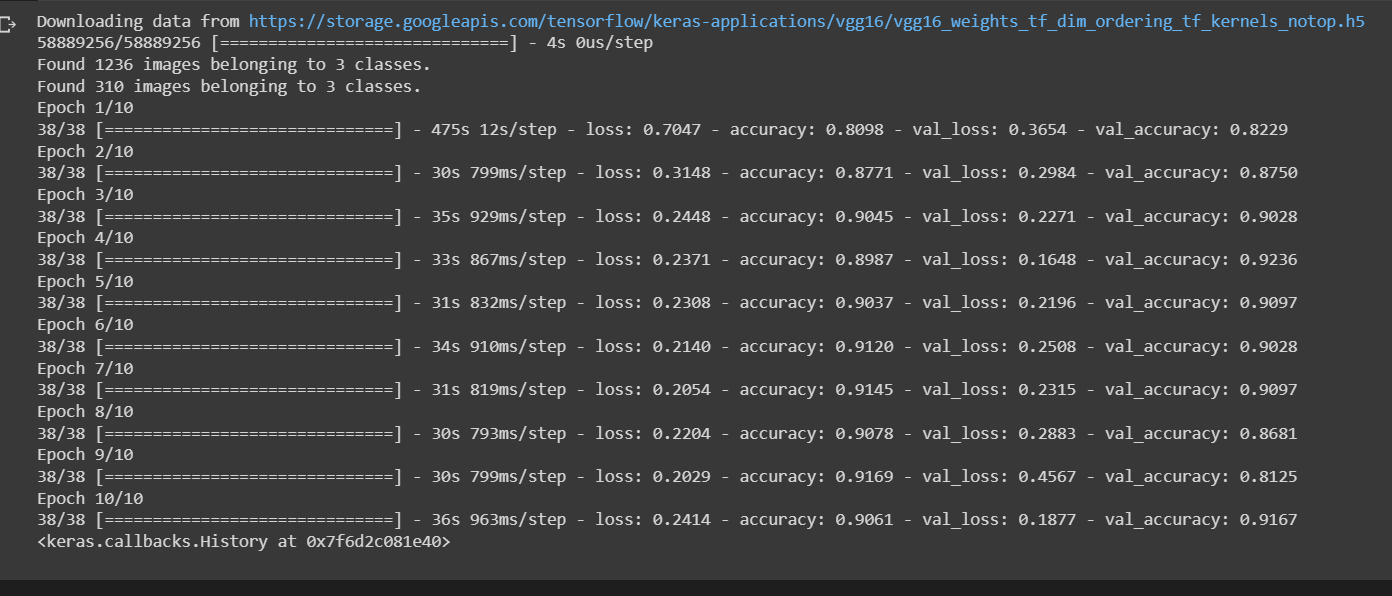
val\_datagen = ImageDataGenerator(rescale=1.0/255.0)

train\_generator = train\_datagen.flow\_from\_directory(train\_dir, target\_size=(img\_width, img\_height), batch\_size=32, class\_mode='categorical')

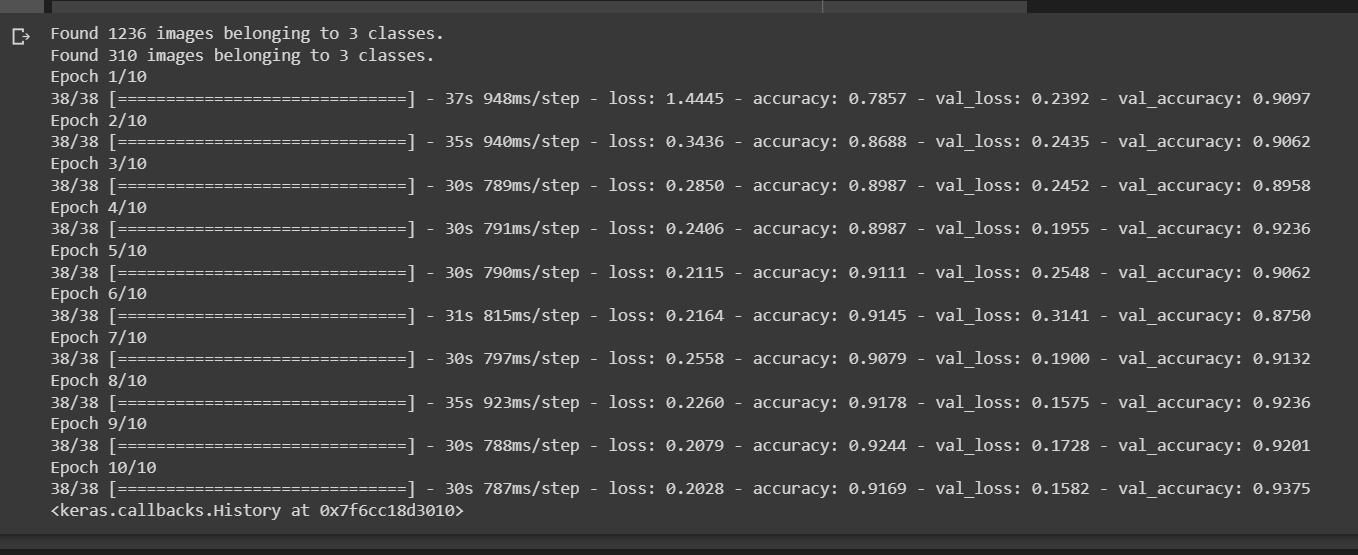
val\_generator = val\_datagen.flow\_from\_directory(val\_dir, target\_size=(img\_width, img\_height), batch\_size=32, class\_mode='categorical')

# Train the model

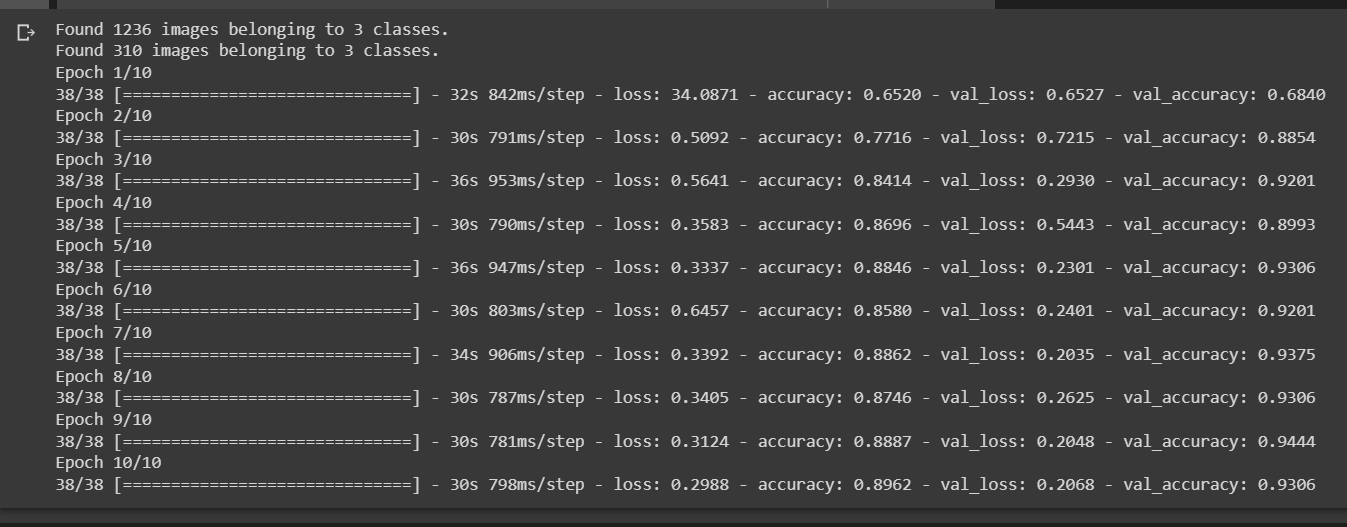
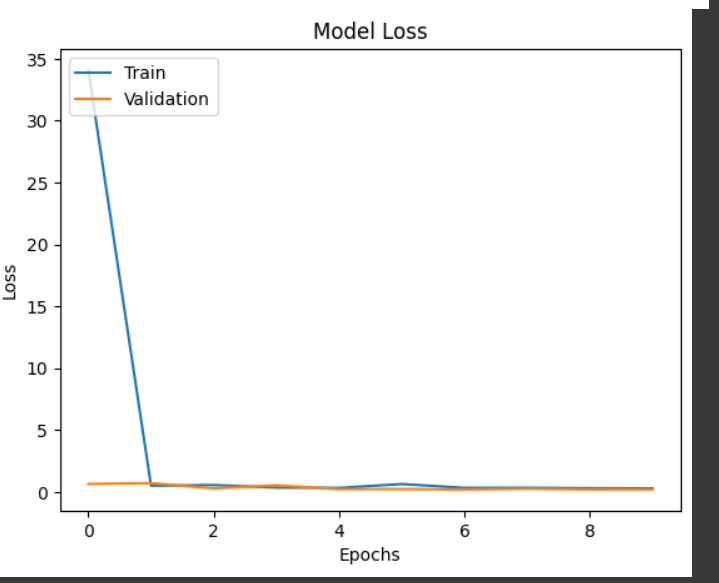
model.fit(train\_generator, steps\_per\_epoch=train\_generator.n // train\_generator.batch\_size, epochs=10, validation\_data=val\_generator, validation\_steps=val\_generator.n // val\_generator.batch\_size)

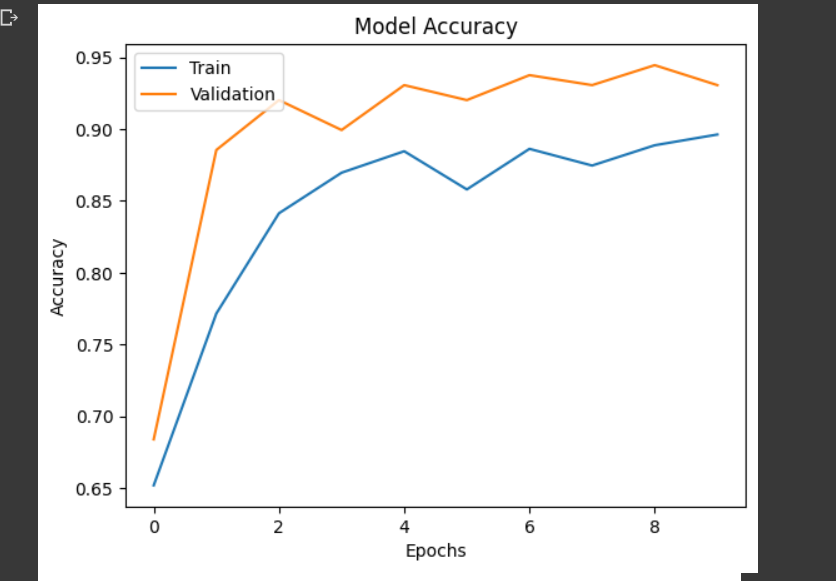


Lr=0.01

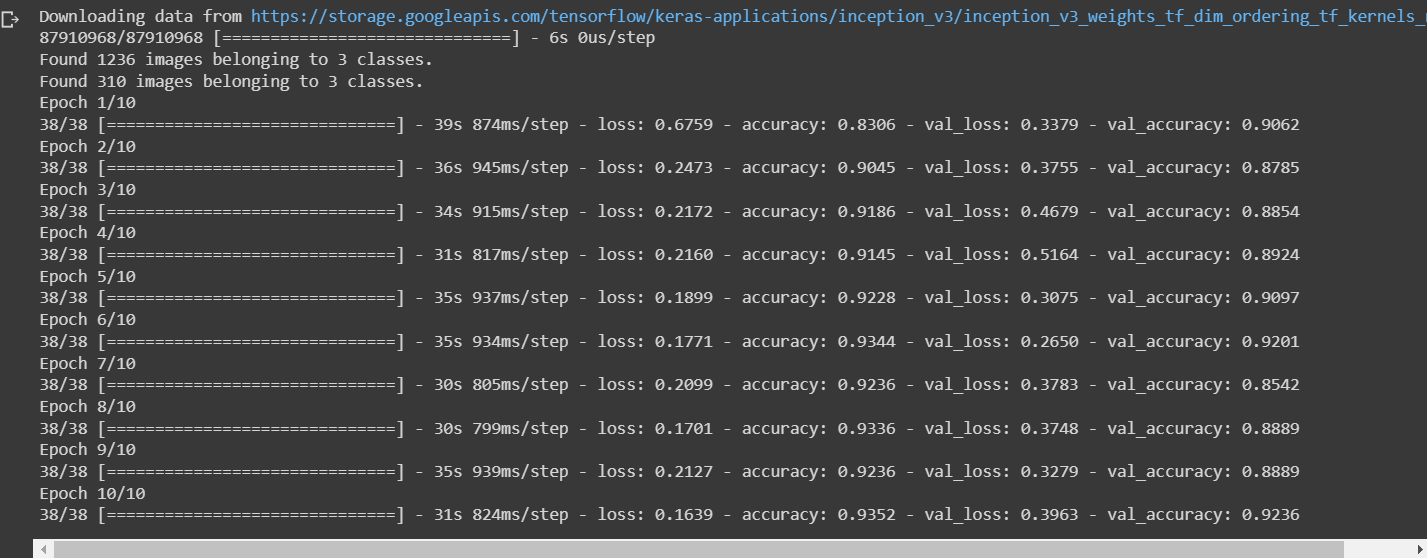


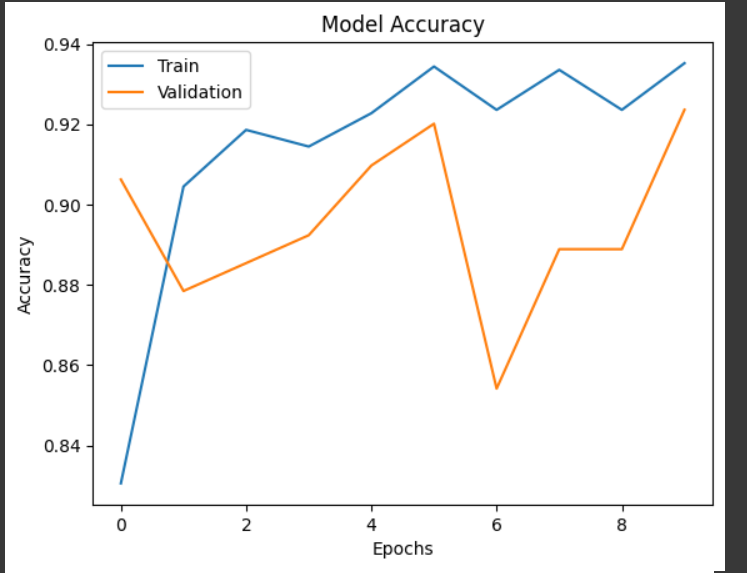
Lr=0.1

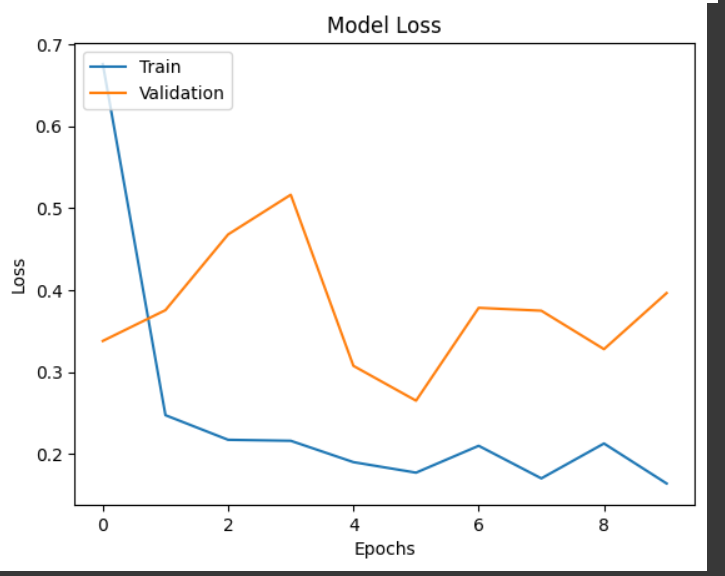
  


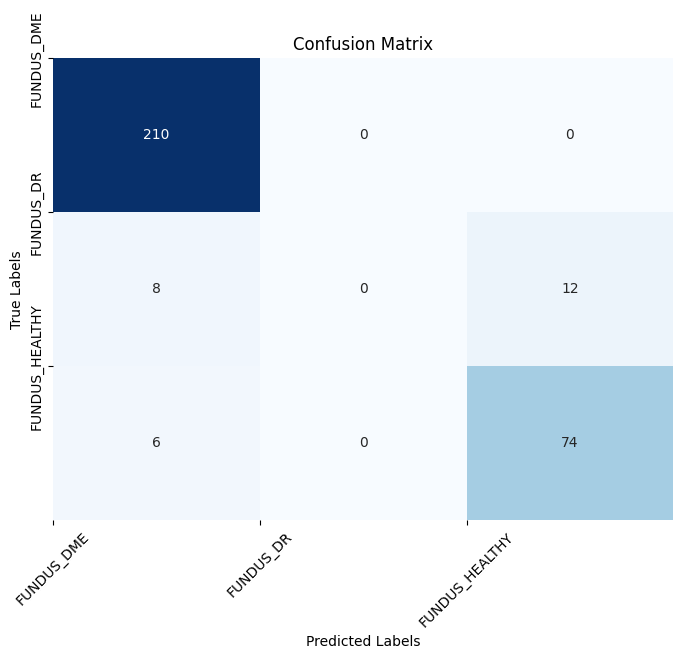


INCEPTIONV3 0.001lr

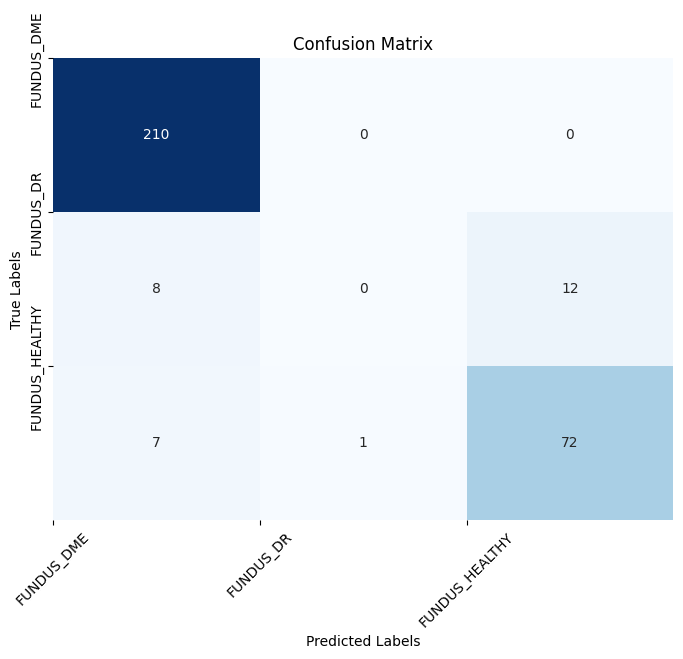


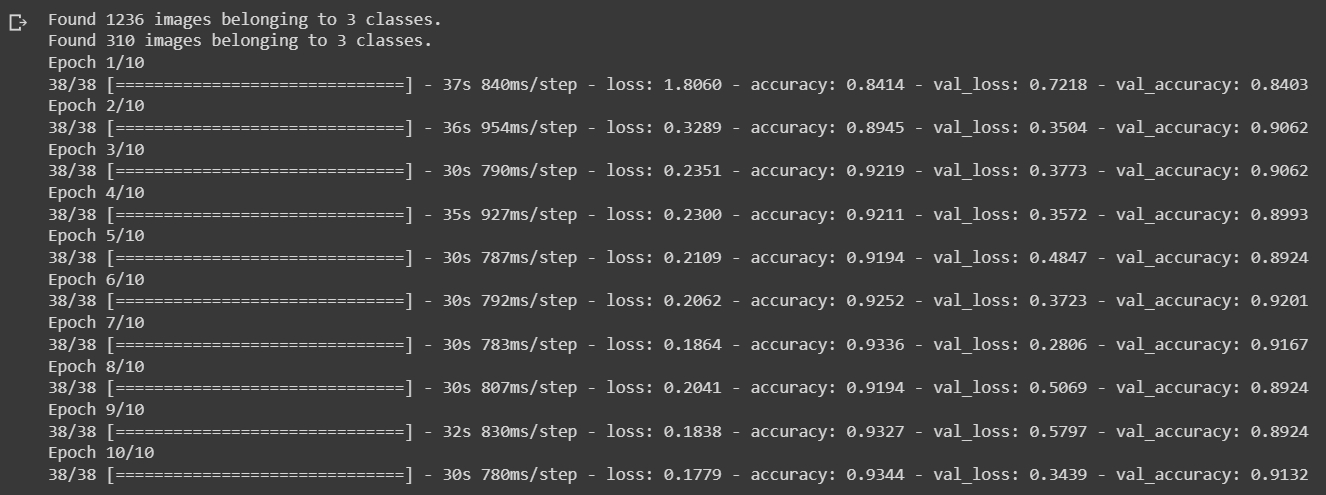


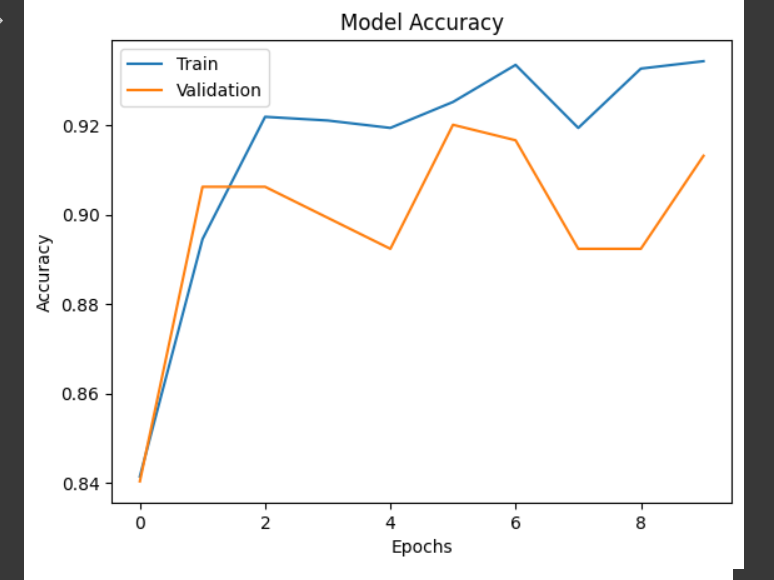


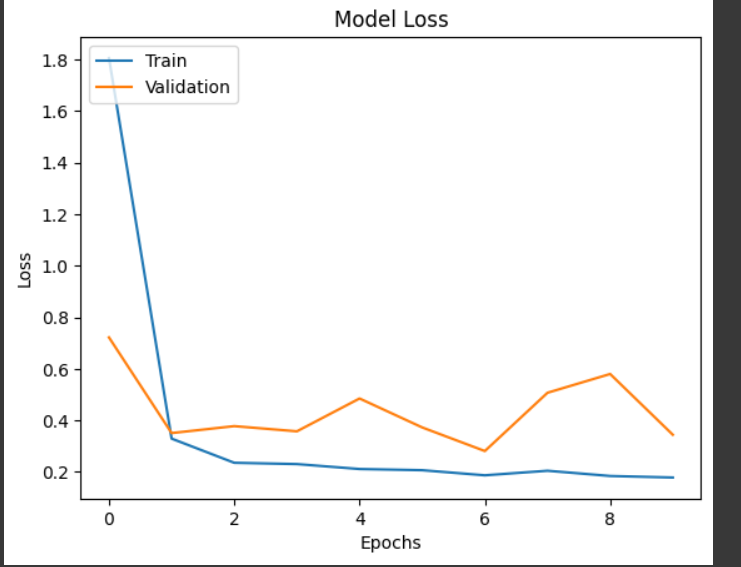


Inception lr=0.01

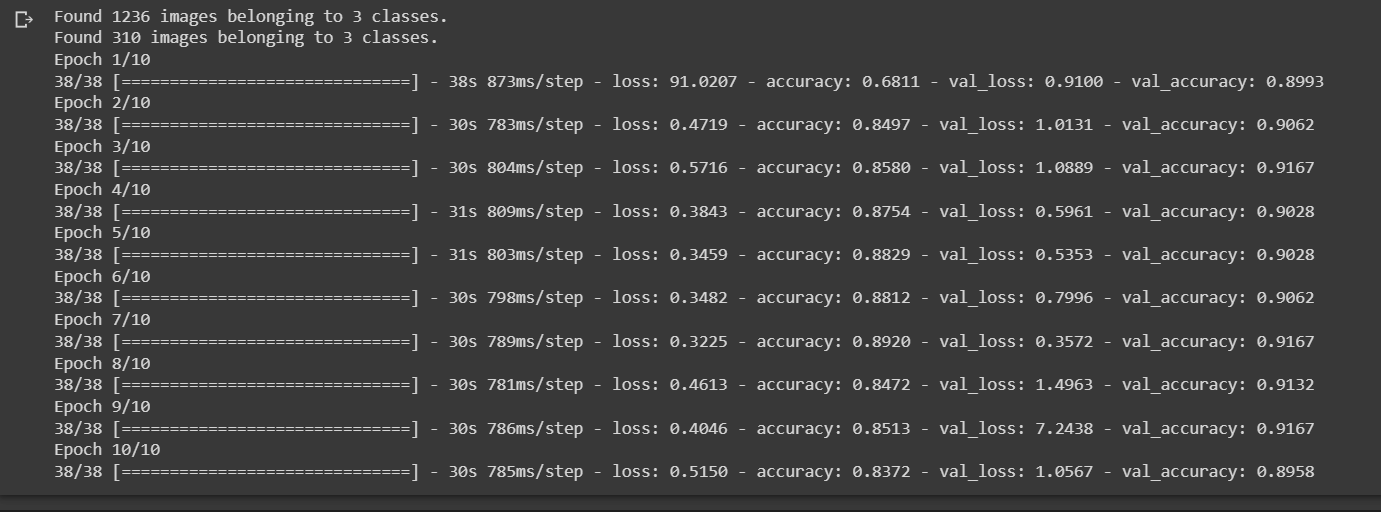


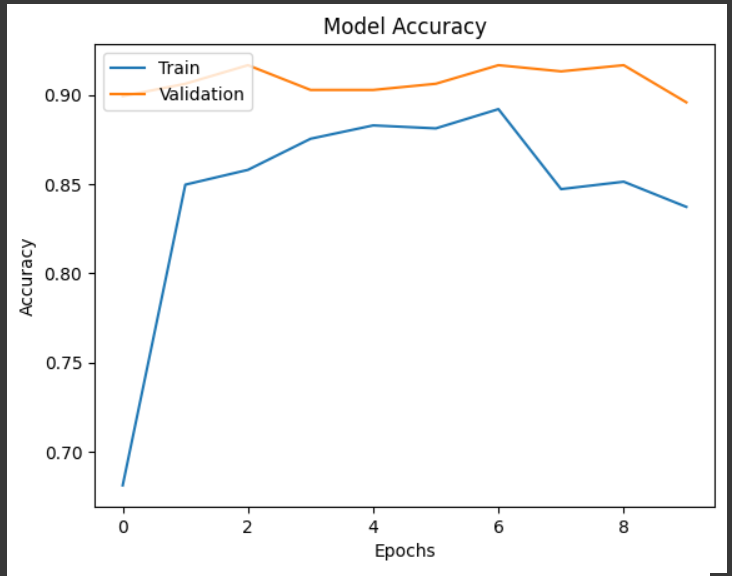


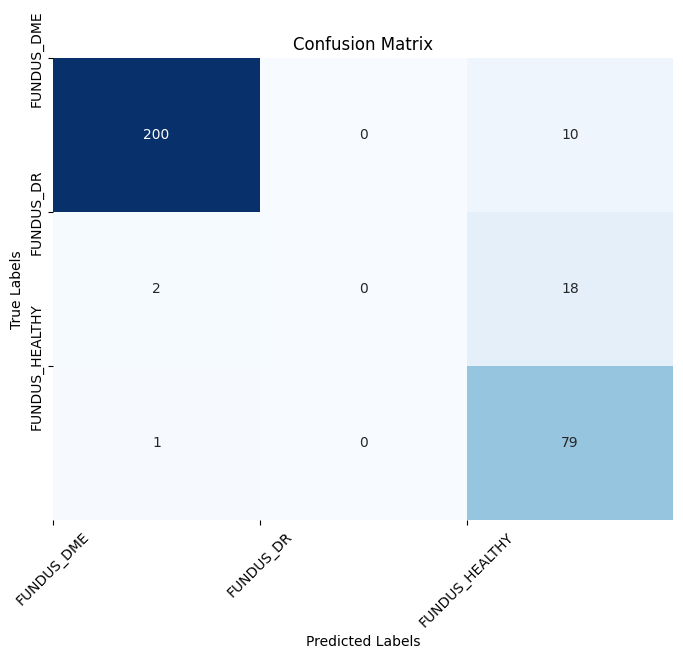


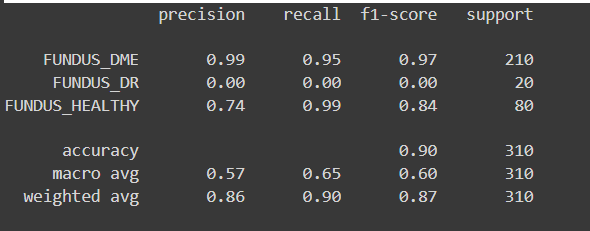


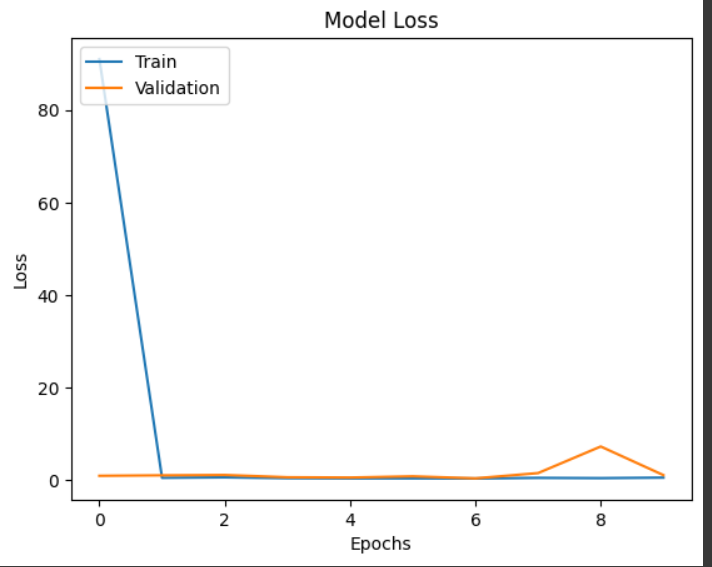
Lr0.1

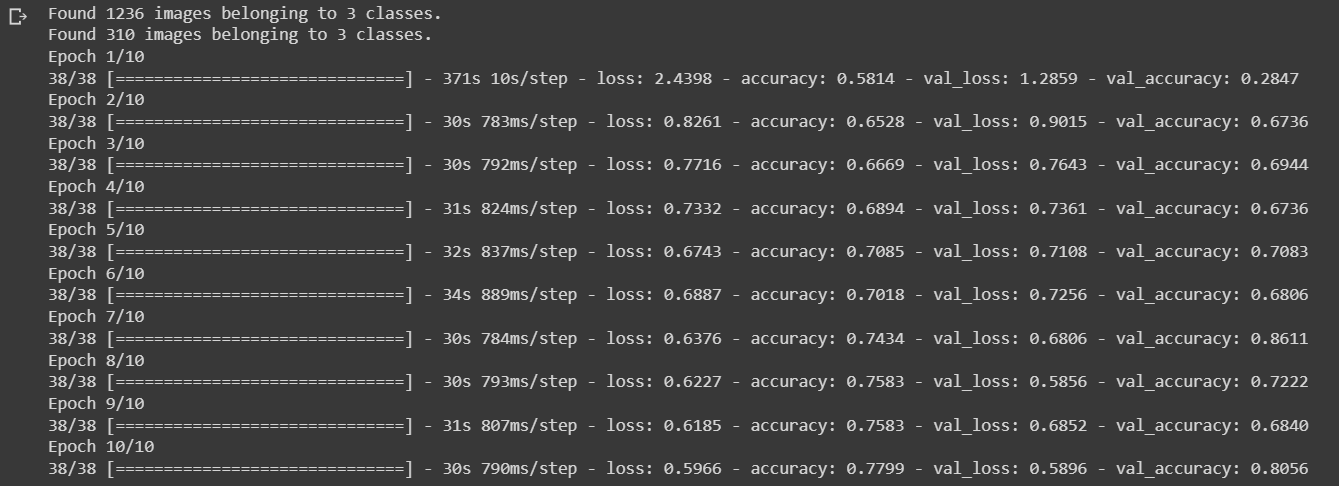


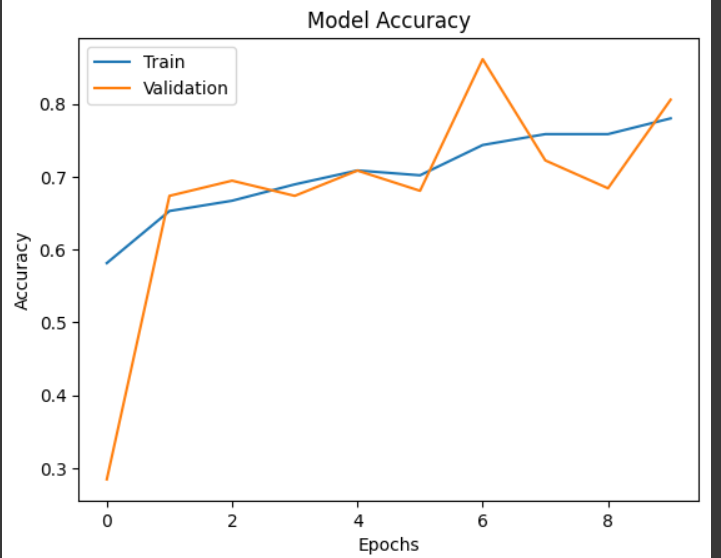


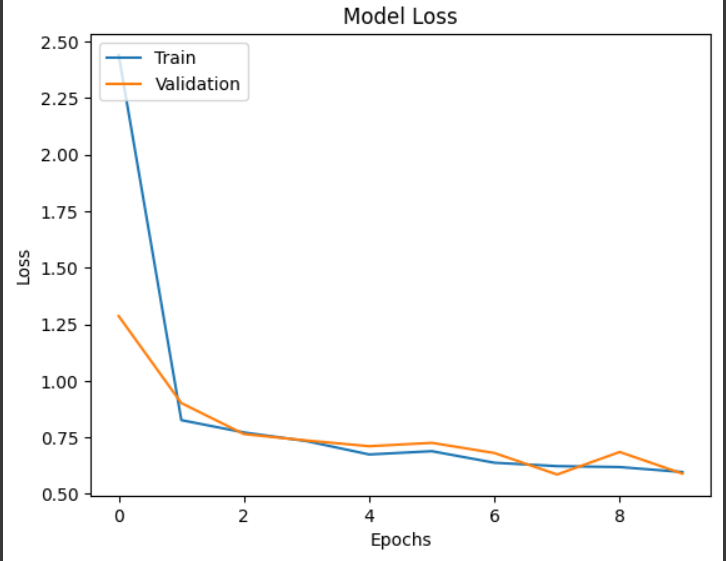


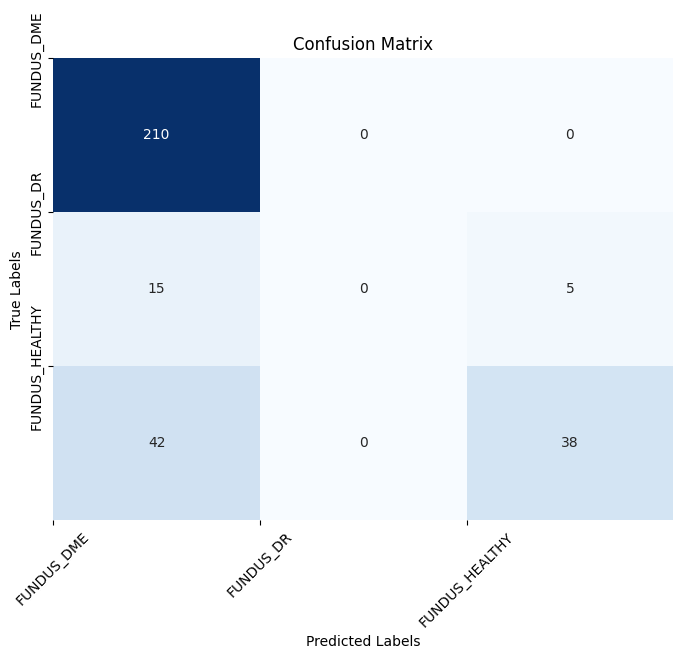


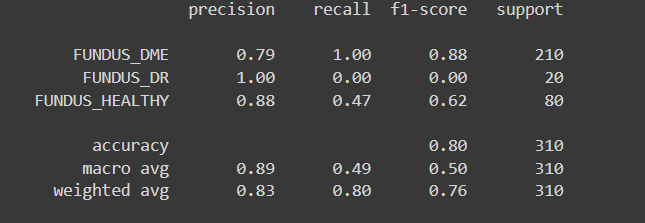
  
resnet50 0.001



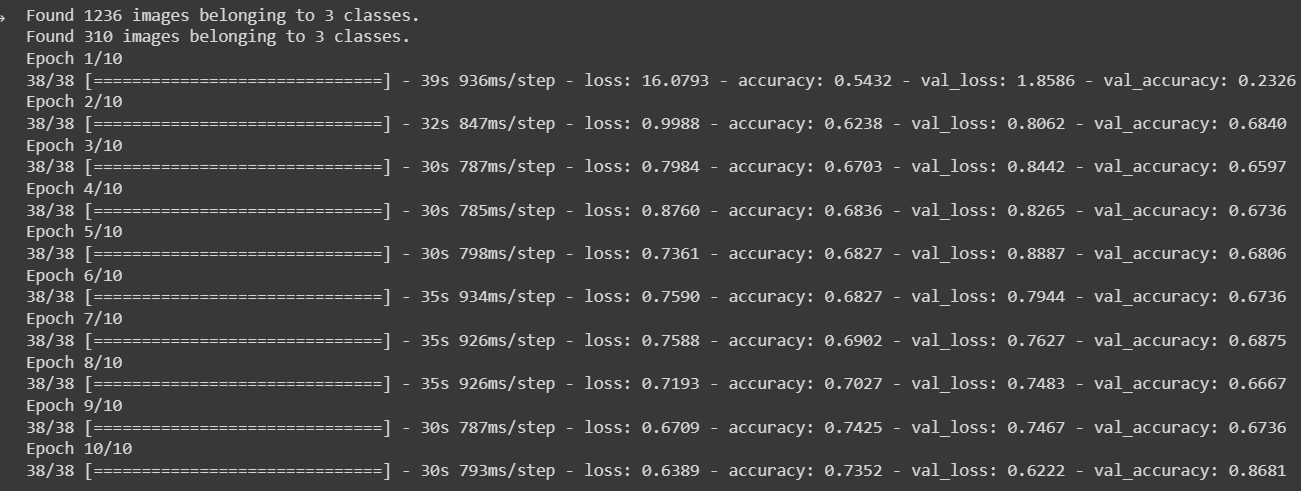


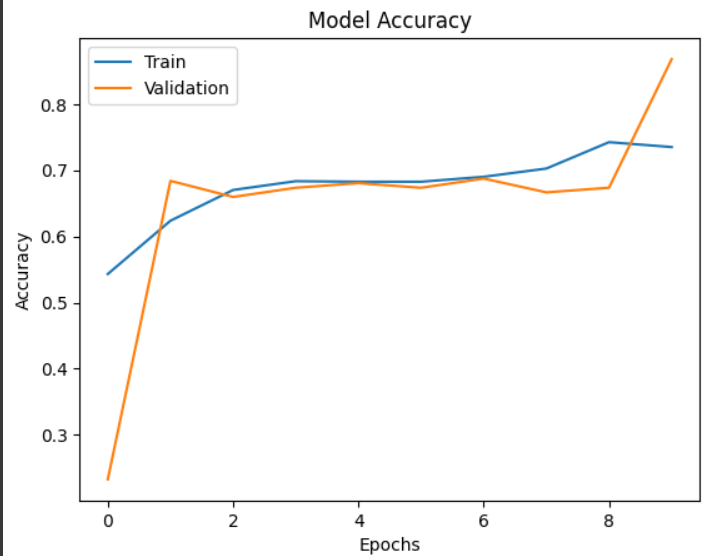


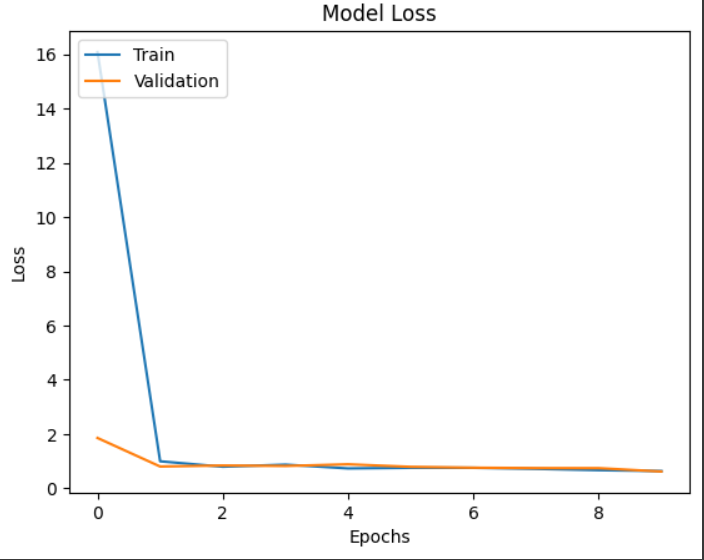


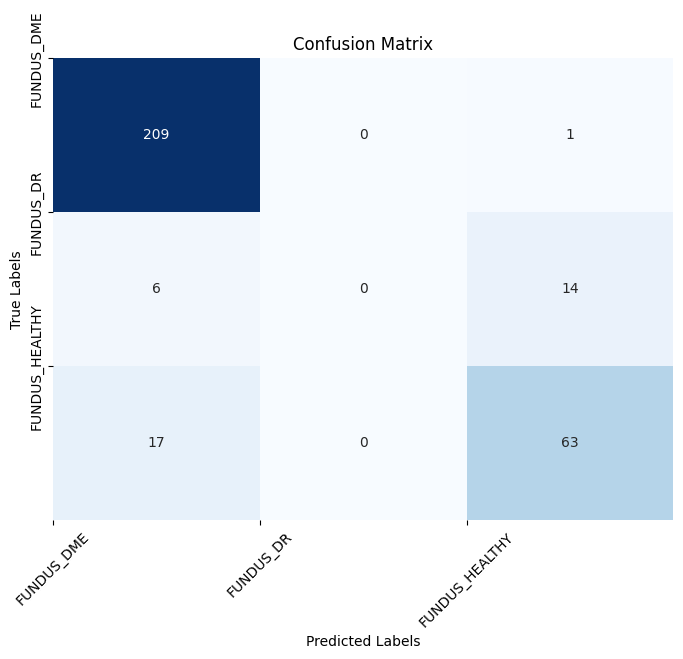


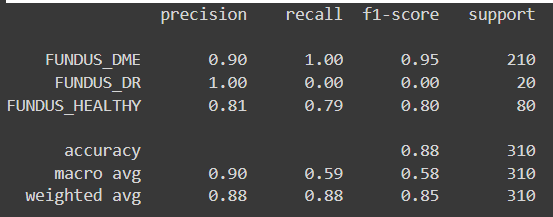
Resnet 0.01











Resnet 0.1

