import os

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import DenseNet121

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.callbacks import ModelCheckpoint

from google.colab import drive

# Mount Google Drive

drive.mount('/content/drive')

# Paths

data\_dir = '/content/drive/MyDrive/embryo\_data'

train\_dir = os.path.join(data\_dir, 'train')

val\_dir = os.path.join(data\_dir, 'val')

test\_dir = os.path.join(data\_dir, 'test')

model\_save\_path = '/content/drive/MyDrive/embryo\_model\_densenet.h5'

# Image parameters

IMG\_SIZE = (224, 224)

BATCH\_SIZE = 32

# Data Generators

train\_datagen = ImageDataGenerator(rescale=1./255,

rotation\_range=20,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

horizontal\_flip=True)

val\_datagen = ImageDataGenerator(rescale=1./255)

train\_generator = train\_datagen.flow\_from\_directory(

train\_dir,

target\_size=IMG\_SIZE,

batch\_size=BATCH\_SIZE,

class\_mode='categorical'

)

val\_generator = val\_datagen.flow\_from\_directory(

val\_dir,

target\_size=IMG\_SIZE,

batch\_size=BATCH\_SIZE,

class\_mode='categorical'

)

# Model building

base\_model = DenseNet121(weights='imagenet', include\_top=False, input\_shape=(224, 224, 3))

x = base\_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(256, activation='relu')(x)

predictions = Dense(train\_generator.num\_classes, activation='softmax')(x)

model = Model(inputs=base\_model.input, outputs=predictions)

# Freeze base model

for layer in base\_model.layers:

layer.trainable = False

# Compile model

model.compile(optimizer=Adam(learning\_rate=1e-4),

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Train model

history = model.fit(

train\_generator,

epochs=10,

validation\_data=val\_generator

)

# Save model

model.save(model\_save\_path)

print(f"Model saved at: {model\_save\_path}")