import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.models import Model

from tensorflow.keras.layers import GlobalAveragePooling2D, Dense

from tensorflow.keras.optimizers import Adam

import os

# Paths & Params

data\_dir = "poultry\_dataset"

img\_size = 224

batch\_size = 32

# 1. Image Preprocessing with Augmentation

datagen = ImageDataGenerator(

rescale=1./255,

rotation\_range=15,

zoom\_range=0.2,

shear\_range=0.1,

horizontal\_flip=True,

validation\_split=0.2

)

train\_data = datagen.flow\_from\_directory(

data\_dir,

target\_size=(img\_size, img\_size),

batch\_size=batch\_size,

class\_mode='categorical',

subset='training'

)

val\_data = datagen.flow\_from\_directory(

data\_dir,

target\_size=(img\_size, img\_size),

batch\_size=batch\_size,

class\_mode='categorical',

subset='validation'

)

# 2. Load Pretrained ResNet50

base\_model = ResNet50(weights='imagenet', include\_top=False, input\_shape=(img\_size, img\_size, 3))

base\_model.trainable = False # Freeze base model

# 3. Add Custom Classification Layers

x = base\_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(128, activation='relu')(x)

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

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

# 4. Compile Model

model.compile(optimizer=Adam(learning\_rate=1e-4), loss='categorical\_crossentropy', metrics=['accuracy'])

# 5. Train Model

history = model.fit(train\_data, validation\_data=val\_data, epochs=10)

# 6. Save the Model

os.makedirs("model", exist\_ok=True)

model.save("model/poultry\_model\_resnet50.h5")

# 7. Evaluate Model

loss, accuracy = model.evaluate(val\_data)

print(f"Validation Accuracy: {accuracy \* 100:.2f}%")