

```

import os
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models

"data_path = "your_dataset_folder
"test_folder_path = "path_to_your_new_images_folder
img_height = 224
img_width = 224
batch_size = 32

)train_ds = tf.keras.utils.image_dataset_from_directory
,data_path
,validation_split=0.2
,"subset="training
,seed=123
,(image_size=(img_height, img_width
batch_size=batch_size
(

)test_ds = tf.keras.utils.image_dataset_from_directory
,data_path
,validation_split=0.2
,"subset="validation
,seed=123
,(image_size=(img_height, img_width
batch_size=batch_size
(

class_names = train_ds.class_names

])augment = models.Sequential
,("layers.RandomFlip("horizontal
,(layers.RandomRotation(0.1
,(layers.RandomZoom(0.1
([

])model = models.Sequential
,((layers.Input(shape=(img_height, img_width, 3
,augment
,(layers.Rescaling(1./255
,('layers.Conv2D(32, 3, activation='relu
,())layers.MaxPooling2D
,('layers.Conv2D(64, 3, activation='relu
,())layers.MaxPooling2D
,())layers.Flatten
,('layers.Dense(128, activation='relu
('layers.Dense(len(class_names), activation='softmax

```

```

([
    model.compile
    ,optimizer='adam
    ,loss='sparse_categorical_crossentropy
    ['metrics=['accuracy
    (

(model.fit(train_ds, validation_data=test_ds, epochs=10

("model.save("my_model.h5

:(if os.path.exists(test_folder_path
:(for filename in os.listdir(test_folder_path
:("if filename.lower().endswith((".jpg", ".png", ".jpeg
(img_path = os.path.join(test_folder_path, filename
img = tf.keras.utils.load_img(img_path, target_size=(img_height,
((img_width
(img_array = tf.keras.utils.img_to_array(img
(img_array = tf.expand_dims(img_array, 0

(predictions = model.predict(img_array, verbose=0
([score = tf.nn.softmax(predictions[0

[(result_label = class_names[np.argmax(score
(confidence = 100 * np.max(score

("%o{print(f"{filename} | {result_label} | {confidence:.2f

```