Import necessary libraries

From tensorflow.keras.models import Sequential

From tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

From tensorflow.keras.preprocessing.image import ImageDataGenerator

From sklearn.metrics import precision\_score, recall\_score, f1\_score

Define the model architecture

Model = Sequential()

Model.add(Conv2D(32, (3, 3), activation=’relu’, input\_shape=(224, 224, 3)))

Model.add(MaxPooling2D((2, 2)))

Model.add(Conv2D(64, (3, 3), activation=’relu’))

Model.add(MaxPooling2D((2, 2)))

Model.add(Conv2D(128, (3, 3), activation=’relu’))

Model.add(MaxPooling2D((2, 2)))

Model.add(Flatten())

Model.add(Dense(128, activation=’relu’))

Model.add(Dense(10, activation=’softmax’))

Compile the model

Model.compile(optimizer=’adam’, loss=’categorical\_crossentropy’, metrics=[‘accuracy’])

Define the data generator

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

Shear\_range=0.2,

Zoom\_range=0.2,

Horizontal\_flip=True)

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

Load the dataset

Train\_generator = train\_datagen.flow\_from\_directory(‘path/to/train/directory’,

Target\_size=(224, 224),

Batch\_size=32,

Class\_mode=’categorical’)

Test\_generator = test\_datagen.flow\_from\_directory(‘path/to/test/directory’,

Target\_size=(224, 224),

Batch\_size=32,

Class\_mode=’categorical’)

Train the model

History = model.fit(train\_generator,

Epochs=10,

Validation\_data=test\_generator)

Evaluate the model

Loss, accuracy = model.evaluate(test\_generator)

Print(f’Test accuracy: {accuracy:.2f}’)

Make predictions

Predictions = model.predict(test\_generator)

Evaluate the model using precision, recall, and F1-score

Precision = precision\_score(test\_generator.classes, np.argmax(predictions, axis=1), average=’macro’)

Recall = recall\_score(test\_generator.classes, np.argmax(predictions, axis=1), average=’macro’)

F1 = f1\_score(test\_generator.classes, np.argmax(predictions, axis=1), average=’macro’)

Print(f’Precision: {precision:.2f}’)

Print(f’Recall: {recall:.2f}’)

Print(f’F1-score: {f1:.2f}’)