import cv2

import numpy as np

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

from keras.utils import to\_categorical

# Load and preprocess the MNIST dataset

(train\_images, train\_labels), (test\_images, test\_labels) = mnist.load\_data()

train\_images = np.expand\_dims(train\_images, axis=-1).astype('float32') / 255

test\_images = np.expand\_dims(test\_images, axis=-1).astype('float32') / 255

train\_labels = to\_categorical(train\_labels)

test\_labels = to\_categorical(test\_labels)

# Build the CNN model

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),

MaxPooling2D((2, 2)),

Flatten(),

Dense(128, activation='relu'),

Dense(10, activation='softmax')

])

# Compile the model

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(train\_images, train\_labels, epochs=5, batch\_size=64, validation\_data=(test\_images, test\_labels))

# Evaluate the model

test\_loss, test\_acc = model.evaluate(test\_images, test\_labels)

print(f"Test accuracy: {test\_acc \* 100:.2f}%")

**Expected Output**

Upon training, the model will display progress for each epoch, and after completion, it will output the test accuracy:

yaml

CopyEdit

Epoch 1/5

938/938 [==============================] - 8s 7ms/step - loss: 0.2969 - accuracy: 0.9172 - val\_loss: 0.1569 - val\_accuracy: 0.9547

Epoch 2/5

938/938 [==============================] - 10s 11ms/step - loss: 0.1352 - accuracy: 0.9602 - val\_loss: 0.1116 - val\_accuracy: 0.9673

...

Test accuracy: 97.50%