import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.datasets import fashion\_mnist

from tensorflow.keras.utils import to\_categorical

# Load and preprocess the Fashion MNIST dataset

(x\_train, y\_train), (x\_test, y\_test) = fashion\_mnist.load\_data()

# Reshape and normalize the data

x\_train = x\_train.reshape((x\_train.shape[0], 28, 28, 1)).astype('float32') / 255

x\_test = x\_test.reshape((x\_test.shape[0], 28, 28, 1)).astype('float32') / 255

# One-hot encode the labels

y\_train = to\_categorical(y\_train)

y\_test = to\_categorical(y\_test)

# Build the CNN model

model = models.Sequential()

# First Convolutional Layer

model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)))

model.add(layers.MaxPooling2D((2, 2)))

# Second Convolutional Layer

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

model.add(layers.MaxPooling2D((2, 2)))

# Third Convolutional Layer

model.add(layers.Conv2D(64, (3, 3), activation='relu'))

# Flatten the output from the previous layer

model.add(layers.Flatten())

# Fully Connected Layer

model.add(layers.Dense(64, activation='relu'))

# Output Layer

model.add(layers.Dense(10, activation='softmax')) # 10 classes for Fashion MNIST

# Compile the model

model.compile(optimizer='adam',

loss='categorical\_crossentropy',

metrics=['accuracy'])

# Train the model

model.fit(x\_train, y\_train, epochs=5, batch\_size=64, validation\_data=(x\_test, y\_test))

# Evaluate the model on the test dataset

test\_loss, test\_acc = model.evaluate(x\_test, y\_test)

print('Test accuracy:', test\_acc)

# Make predictions on new data

predictions = model.predict(x\_test)

print('Predictions:', predictions[0])