import numpy as np

import tensorflow as tf

# Step 1: Define the 5x5 input matrix

input\_matrix = np.array([[1, 2, 3, 4, 5],

[6, 7, 8, 9, 10],

[11, 12, 13, 14, 15],

[16, 17, 18, 19, 20],

[21, 22, 23, 24, 25]], dtype=np.float32) # Cast to float32

# Step 2: Define the 3x3 kernel with the provided values

kernel = np.array([[0, 1, 0],

[1, -4, 1],

[0, 1, 0]], dtype=np.float32) # Cast to float32

# Reshape the input matrix and kernel for TensorFlow convolution

input\_tensor = np.expand\_dims(np.expand\_dims(input\_matrix, axis=-1), axis=0) # Shape: (1, 5, 5, 1)

kernel\_tensor = np.expand\_dims(np.expand\_dims(kernel, axis=-1), axis=-1) # Shape: (3, 3, 1, 1)

# Step 3: Perform convolution with different parameters

# Stride = 1, Padding = 'VALID'

conv\_valid\_stride\_1 = tf.nn.conv2d(input\_tensor, kernel\_tensor, strides=[1, 1, 1, 1], padding='VALID')

# Stride = 1, Padding = 'SAME'

conv\_same\_stride\_1 = tf.nn.conv2d(input\_tensor, kernel\_tensor, strides=[1, 1, 1, 1], padding='SAME')

# Stride = 2, Padding = 'VALID'

conv\_valid\_stride\_2 = tf.nn.conv2d(input\_tensor, kernel\_tensor, strides=[1, 2, 2, 1], padding='VALID')

# Stride = 2, Padding = 'SAME'

conv\_same\_stride\_2 = tf.nn.conv2d(input\_tensor, kernel\_tensor, strides=[1, 2, 2, 1], padding='SAME')

# Step 4: Print the output feature maps for each case

print("Output Feature Map (Stride = 1, Padding = 'VALID'):")

print(conv\_valid\_stride\_1.numpy().squeeze())

print("\nOutput Feature Map (Stride = 1, Padding = 'SAME'):")

print(conv\_same\_stride\_1.numpy().squeeze())

print("\nOutput Feature Map (Stride = 2, Padding = 'VALID'):")

print(conv\_valid\_stride\_2.numpy().squeeze())

print("\nOutput Feature Map (Stride = 2, Padding = 'SAME'):")

print(conv\_same\_stride\_2.numpy().squeeze())