Illustrate correlation and convolution operation

Using filter2D & convolve function

Apply the following kernels

1. Identity kernel
2. Edge detection kernel
3. Sharpen Kernel
4. Box Blur Kernel
5. Gaussian Blur Kernel

Apply Convolve function in SCIPY library and use the following mode ‘reflect’, ‘constant’, ‘nearest’, ‘mirror’, ‘wrap'

In [4]:

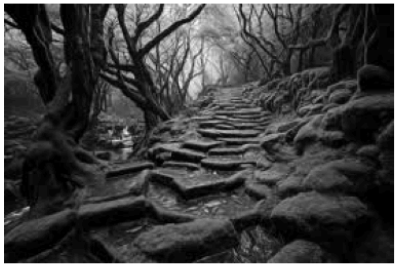
```
import numpy as np
import cv2
import scipy.ndimage
import matplotlib.pyplot as plt

# Load the high-resolution image and convert to grayscale
image_path = 'img1.jpg'
image = cv2.imread(image_path)
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

plt.imshow(gray_image, cmap='gray')
plt.title("Grayscale Image")
plt.axis('off')
plt.show()
```

In [5]:

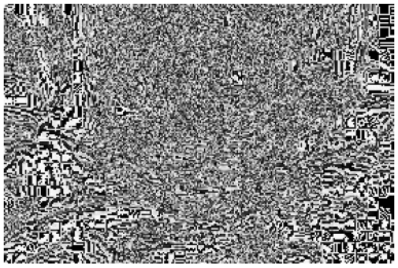
```
# Define the kernels
identity_kernel = np.array([[0, 0, 0], [0, 1, 0], [0, 0, 0]])
edge_detection_kernel = np.array([[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]])
sharpen_kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
box_blur_kernel = np.ones((3, 3), np.float32) / 9.0
gaussian_blur_kernel = np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]]) / 16.0

# List of kernels for easy iteration
kernels = {
    'Identity': identity_kernel,
    'Edge Detection': edge_detection_kernel,
    'Sharpen': sharpen_kernel,
    'Box Blur': box_blur_kernel,
    'Gaussian Blur': gaussian_blur_kernel
}

# List of boundary modes
modes = ['reflect', 'constant', 'nearest', 'mirror', 'wrap']

# Apply the SciPy convolve function to each kernel and boundary mode
def apply_convolution(image, kernel, mode):
    return scipy.ndimage.convolve(image, kernel, mode=mode)

# Process each kernel with each mode
for mode in modes:
    print(f"\nApplying kernels with SciPy's convolve in '{mode}' mode:")
    plt.figure(figsize=(12, 8))
    for i, (name, kernel) in enumerate(kernels.items()):
        convolved_image = apply_convolution(gray_image, kernel, mode)
        plt.subplot(2, 3, i+1)
        plt.imshow(convolved_image, cmap='gray')
        plt.title(f"{name} Kernel - {mode} Mode")
        plt.axis('off')
    plt.tight_layout()
    plt.show()
```

```
Applying kernels with SciPy's convolve in 'reflect' mode:
```

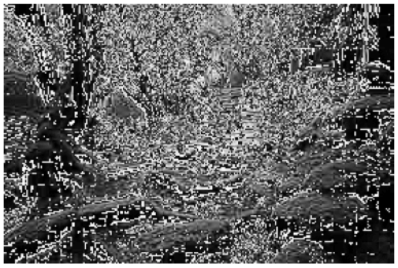
Identity Kernel - reflect Mode



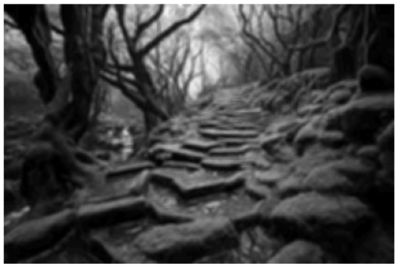
Edge Detection Kernel - reflect Mode



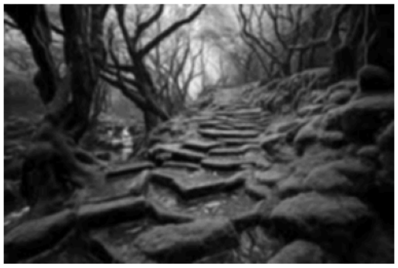
Sharpen Kernel - reflect Mode



Box Blur Kernel - reflect Mode

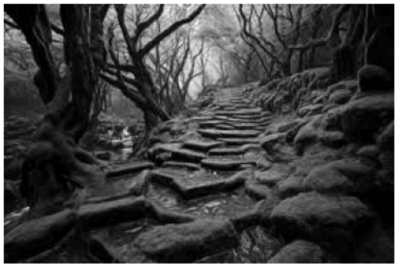


Gaussian Blur Kernel - reflect Mode

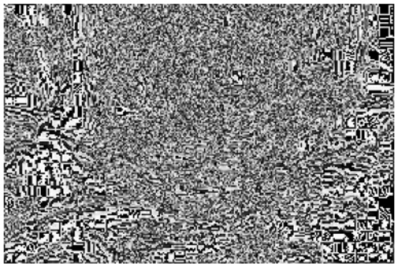


```
Applying kernels with SciPy's convolve in 'constant' mode:
```

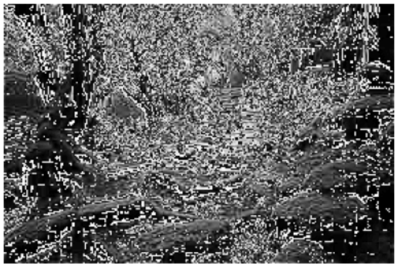
Identity Kernel - constant Mode



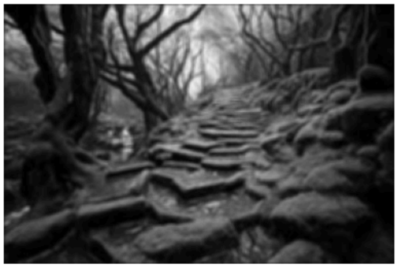
Edge Detection Kernel - constant Mode



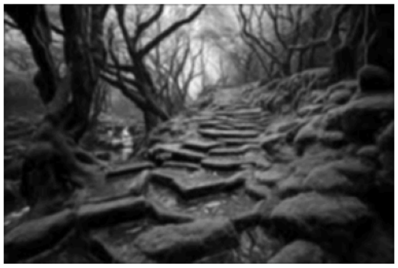
Sharpen Kernel - constant Mode



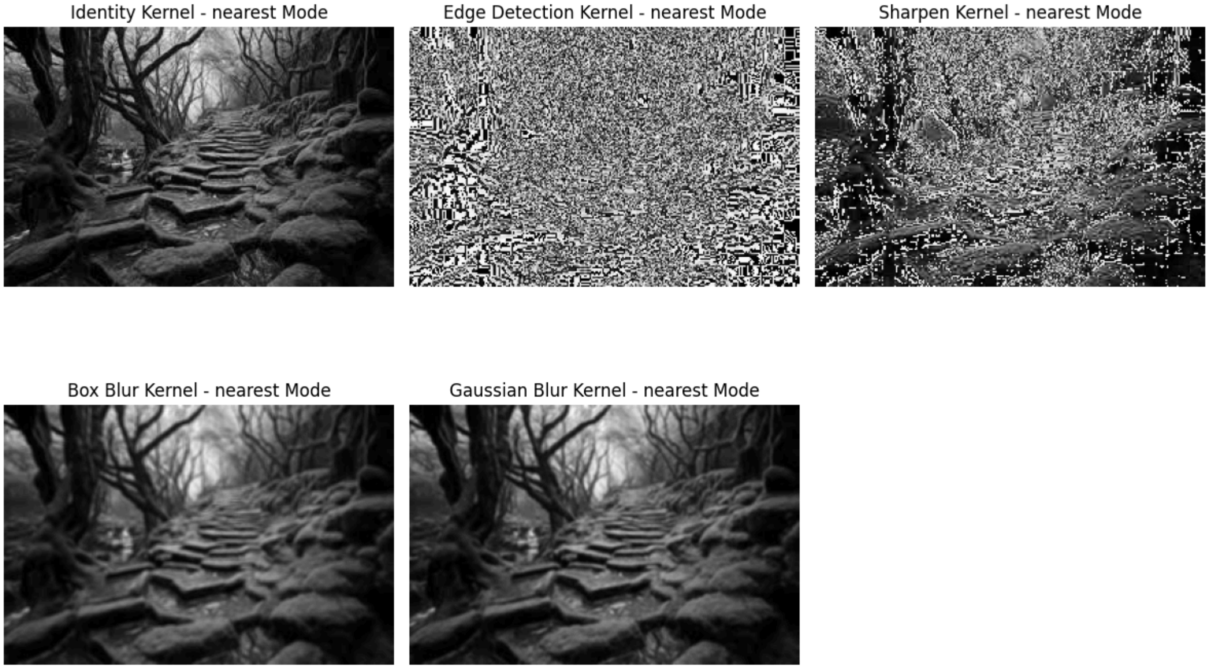
Box Blur Kernel - constant Mode



Gaussian Blur Kernel - constant Mode



```
Applying kernels with SciPy's convolve in 'nearest' mode:
```



Identity Kernel - nearest Mode

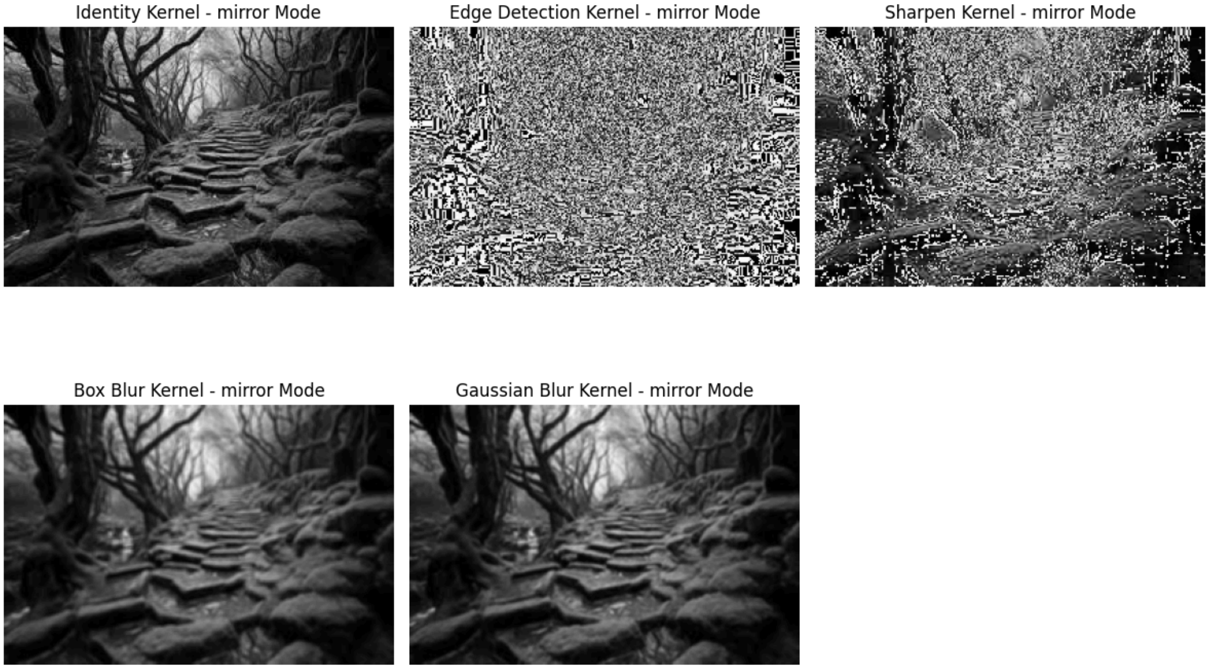
Edge Detection Kernel - nearest Mode

Sharpen Kernel - nearest Mode

Box Blur Kernel - nearest Mode

Gaussian Blur Kernel - nearest Mode

```
Applying kernels with SciPy's convolve in 'mirror' mode:
```



Identity Kernel - mirror Mode

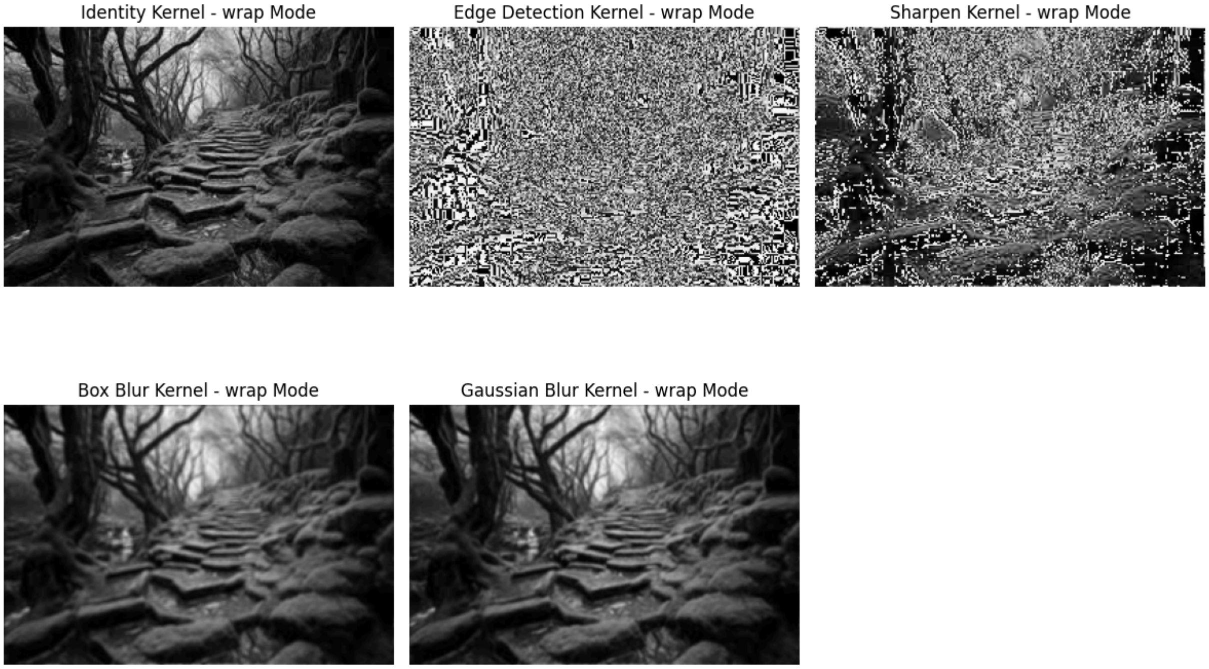
Edge Detection Kernel - mirror Mode

Sharpen Kernel - mirror Mode

Box Blur Kernel - mirror Mode

Gaussian Blur Kernel - mirror Mode

```
Applying kernels with SciPy's convolve in 'wrap' mode:
```



Identity Kernel - wrap Mode

Edge Detection Kernel - wrap Mode

Sharpen Kernel - wrap Mode

Box Blur Kernel - wrap Mode

Gaussian Blur Kernel - wrap Mode