In [ ]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
image = cv2.imread('test_image3.jpg')
```

In [ ]:
```
gaussian_noise = np.random.normal(0, 0.6, image.shape).astype(np.uint8) # where 0 i
gaussian_noisy_image = cv2.add(image, gaussian_noise)

kernel_size = 3  # Define the kernel size
gaussian_denoised_image = cv2.blur(gaussian_noisy_image, (kernel_size, kernel_size)

kernel = np.ones((kernel_size, kernel_size), dtype=np.float32) / (kernel_size * ker
gaussian_denoised_image2 = cv2.filter2D(gaussian_noisy_image, -1, kernel)  # Convol

plt.figure(figsize=(5, 5))

plt.subplot(2, 2, 1)
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(cv2.cvtColor(gaussian_noisy_image, cv2.COLOR_BGR2RGB))
plt.title('Noisy Image')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(cv2.cvtColor(gaussian_denoised_image, cv2.COLOR_BGR2RGB))
plt.title('Denoise Image 1')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(gaussian_denoised_image2, cv2.COLOR_BGR2RGB))
plt.title('Denoise Image 2')
plt.axis('off')

plt.show()
```

In [ ]:
```
image2 = cv2.imread('test_image3.jpg')

sp_noisy_image = np.copy(image2)

num_salt = np.ceil(0.005 * image2.size * 0.5) # number of pixels that will be salt
num_pepper = np.ceil(0.005 * image2.size * 0.5) # number of pixels that will be pep

print(num_salt, num_pepper)
```

In [ ]:
```
# Add salt noise
coords = [np.random.randint(0, i - 1, int(num_salt)) for i in image2.shape]
sp_noisy_image[coords[0], coords[1], :] = 255 # the ':' means the specified coords

# Add pepper noise
```

```
coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in image2.shape]
sp_noisy_image[coords[0], coords[1], :] = 0
```

In [ ]:
```
# Denoising using cv2 median blur function
sp_denoised_image = cv2.medianBlur(sp_noisy_image, 3)  # Apply median filter with k

# Denoising using median filter matrix
rows, cols, channels = sp_noisy_image.shape
sp_denoised_image2 = np.zeros((rows-2, cols-2, channels), dtype=np.uint8)
for i in range(1, rows - 1):
    for j in range(1, cols - 1):
        for k in range(channels):
            window = sp_noisy_image[i-1:i+2, j-1:j+2, k]
            sp_denoised_image2[i-1, j-1, k] = np.median(window)
```

In [ ]:
```
plt.figure(figsize=(5, 5))

plt.subplot(2, 2, 1)
plt.imshow(cv2.cvtColor(image2, cv2.COLOR_BGR2RGB))
plt.title('Original Image')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(cv2.cvtColor(sp_noisy_image, cv2.COLOR_BGR2RGB))
plt.title('Noisy Image')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(cv2.cvtColor(sp_denoised_image, cv2.COLOR_BGR2RGB))
plt.title('Denoised Image')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(sp_denoised_image2, cv2.COLOR_BGR2RGB))
plt.title('Denoised Image 2')
plt.axis('off')

plt.show()
```