

```
In [1]: # ! unzip imgs.zip
```

```
In [2]: import cv2
import matplotlib.pyplot as plt
import os
```

Read image dataset

```
In [3]: folder_path = "imgs/"
```

```
In [4]: image_files = [f for f in os.listdir(folder_path) if f.endswith('.png')]
```

```
In [5]: fig, axes = plt.subplots(3, 3, figsize=(10, 10))
axes = axes.flatten()
for i, ax in enumerate(axes):
    image = cv2.imread(os.path.join(folder_path, image_files[i]))
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    ax.imshow(image)
    ax.axis('off')
    ax.set_title(f'Image {i+1}')

plt.tight_layout()
plt.show()
```

Image 1



Image 2



Image 3



Image 4



Image 5



Image 6



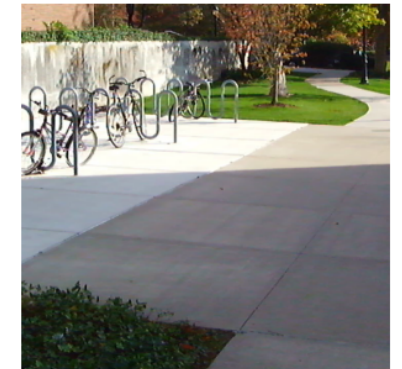
Image 7



Image 8



Image 9



Sobel edge detection

```
In [6]: fig, axes = plt.subplots(3, 3, figsize=(10, 10))
        axes = axes.flatten()
        for i, ax in enumerate(axes):
            image = cv2.imread(os.path.join(folder_path, image_files[i]))

            image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            image_blur = cv2.GaussianBlur(image_gray, (3,3), 0)
            sobelxy = cv2.Sobel(src=image_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize
            sobelxy = cv2.convertScaleAbs(sobelxy)

            ax.imshow(sobelxy, cmap='gray')
            ax.axis('off')
            ax.set_title(f'Image {i+1}')
```

```
plt.tight_layout()
plt.show()
```

Image 1



Image 2



Image 3

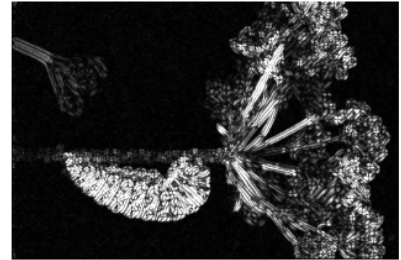


Image 4

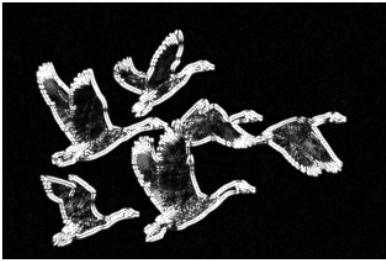


Image 5

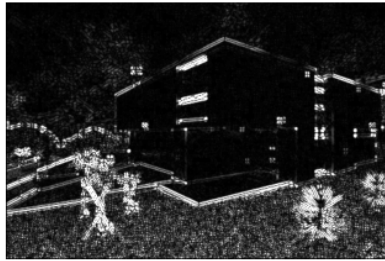


Image 6

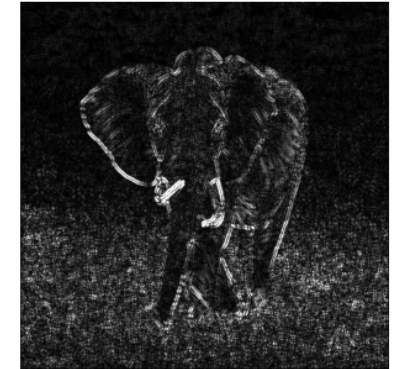


Image 7

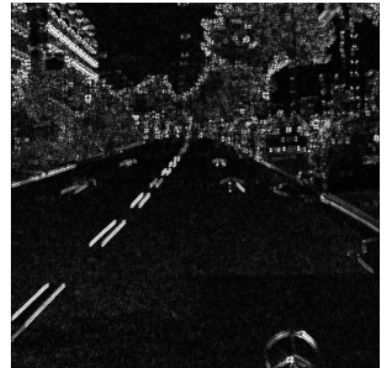
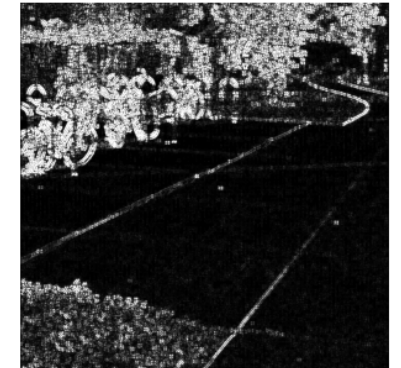


Image 8



Image 9



Canny edge detection

```
In [7]: fig, axes = plt.subplots(3, 3, figsize=(10, 10))
        axes = axes.flatten()
        for i, ax in enumerate(axes):
            image = cv2.imread(os.path.join(folder_path, image_files[i]))

            image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            image_blur = cv2.GaussianBlur(image_gray, (3,3), 0)
            edges = cv2.Canny(image_blur, 50, 50)

            ax.imshow(edges, cmap='gray')
            ax.axis('off')
            ax.set_title(f'Image {i+1}')
```

```
plt.tight_layout()
plt.show()
```

Image 1



Image 2



Image 3



Image 4



Image 5



Image 6



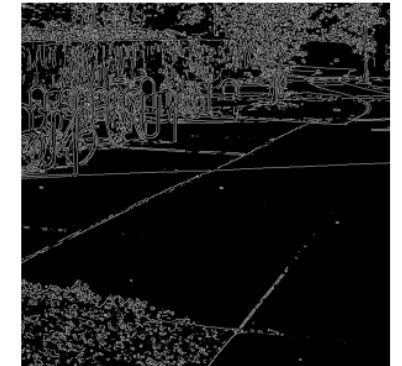
Image 7



Image 8



Image 9



Laplacian edge detection

```
In [8]: fig, axes = plt.subplots(3, 3, figsize=(10, 10))
        axes = axes.flatten()
        for i, ax in enumerate(axes):
            image = cv2.imread(os.path.join(folder_path, image_files[i]))

            image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            image_blur = cv2.GaussianBlur(image_gray, (3,3), 0)
            edges = cv2.Laplacian(image_blur, cv2.CV_64F)
            edges = cv2.convertScaleAbs(edges)

            ax.imshow(edges, cmap='gray')
            ax.axis('off')
            ax.set_title(f'Image {i+1}')
```



```
plt.tight_layout()  
plt.show()
```

Image 1



Image 2

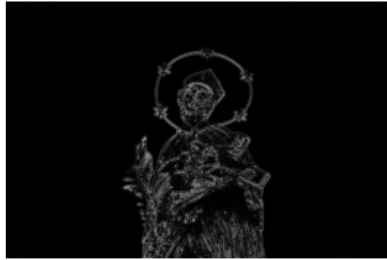


Image 3

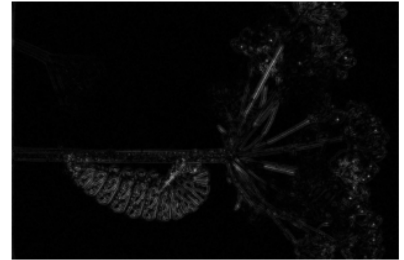


Image 4



Image 5



Image 6



Image 7



Image 8



Image 9

