Practical no 10 : Numpy with picture

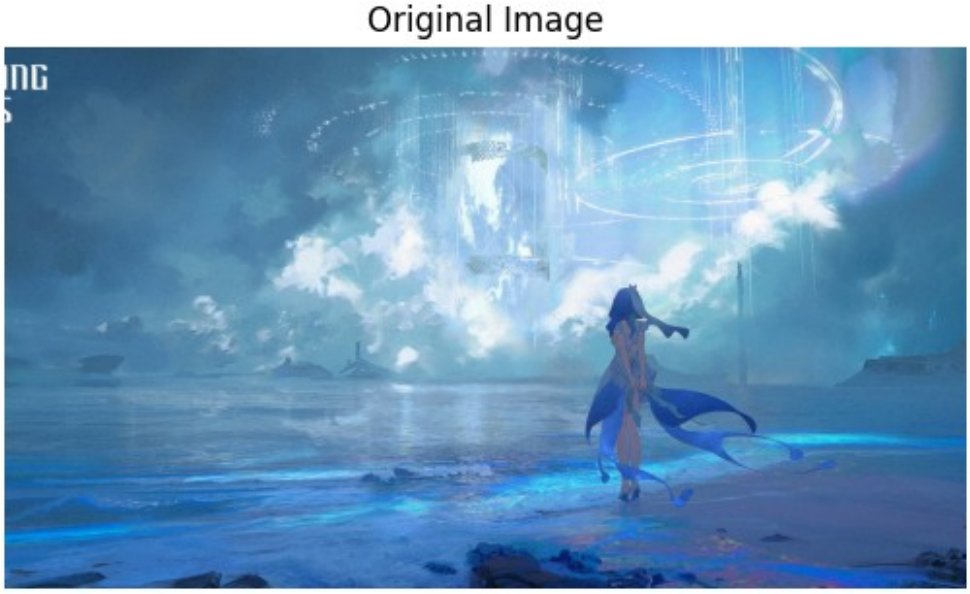
By Parth Gawad, Roll no 62

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

image = cv2.imread('wallhaven-l8633r_1366x768.png')

image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(image_rgb)
plt.title("Original Image")
plt.axis("off")
plt.show()
```



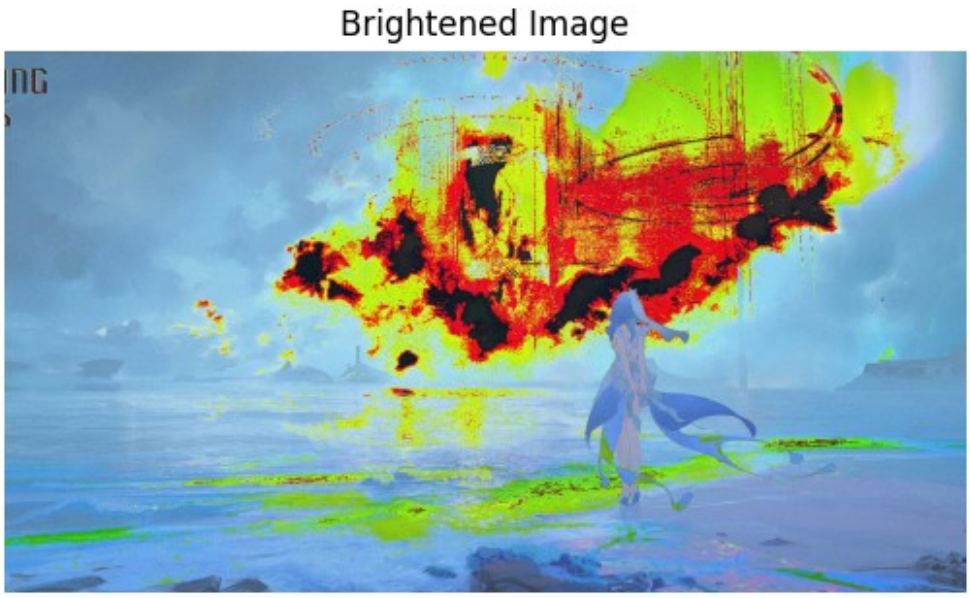

```
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(image_gray, cmap='gray')
plt.title("Grayscale Image")
plt.axis("off")
plt.show()
```

Grayscale Image

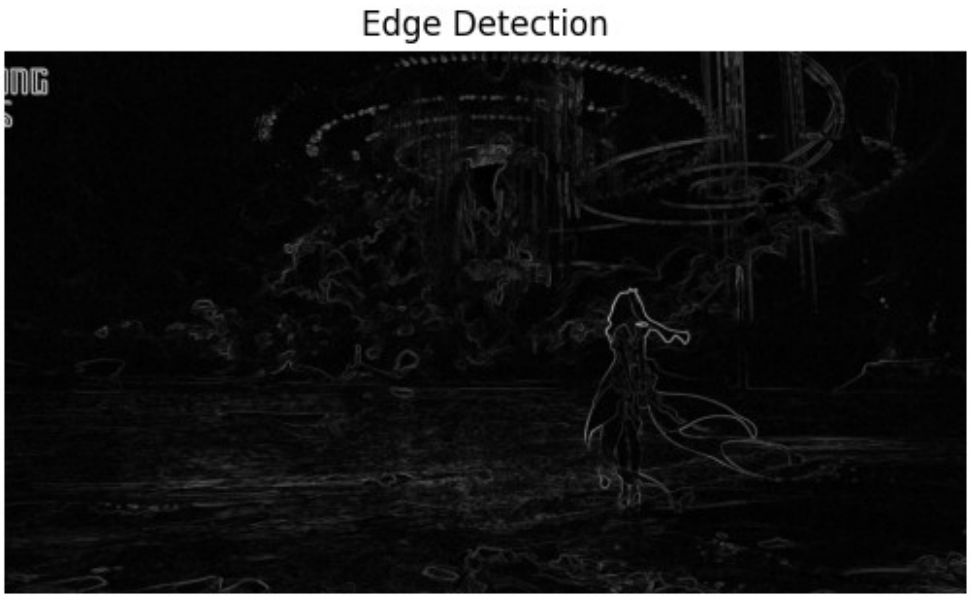
```
cropped_image = image_rgb[50:200, 50:200]
plt.imshow(cropped_image)
plt.title("Cropped Image")
plt.axis("off")
plt.show()
```

Cropped Image

```
bright_image = np.clip(image_rgb + 50, 0, 255)
plt.imshow(bright_image.astype(np.uint8))
plt.title("Brightened Image")
plt.axis("off")
plt.show()
```




```
sobel_x = cv2.Sobel(image_gray, cv2.CV_64F, 1, 0, ksize=5)
sobel_y = cv2.Sobel(image_gray, cv2.CV_64F, 0, 1, ksize=5)
edge_image = np.sqrt(sobel_x**2 + sobel_y**2)
plt.imshow(edge_image, cmap='gray')
plt.title("Edge Detection")
plt.axis("off")
plt.show()
```




```
cv2.imwrite('modified_image.jpg', cv2.cvtColor(bright_image,
cv2.COLOR_RGB2BGR))
```

```
True
```