

Experiment10

April 1, 2025

Name - Shravani Sandeep Raut

SE - 48

```
[19]: # pip install numpy
      # pip install matplotlib
      # pip install opencv-python
```

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

```
[11]: image = cv2.imread('new.jpg')
```

```
[12]: image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

```
[13]: plt.imshow(image_rgb)
     plt.title("Original Image")
     plt.axis("off")
     plt.show()
```

Original Image



```
[14]: 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



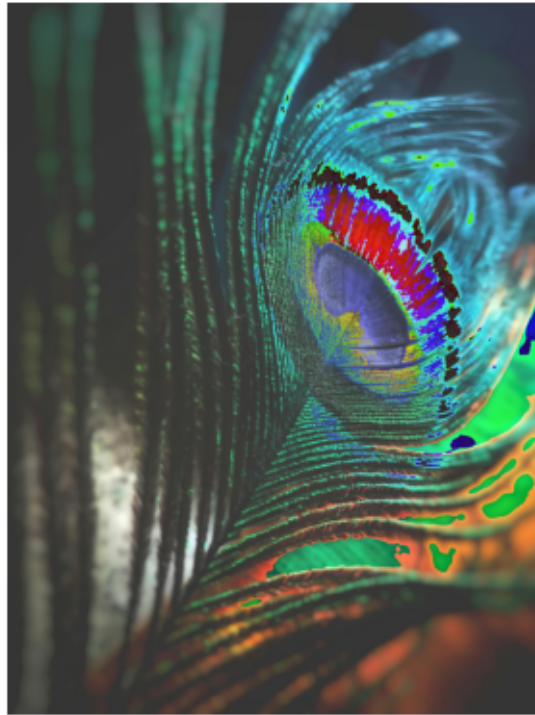
```
[15]: cropped_image = image_rgb[50:200, 50:200]
plt.imshow(cropped_image)
plt.title("Cropped Image")
plt.axis("off")
plt.show()
```

Cropped Image



```
[16]: 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()
```

Brightened Image



```
[17]: 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()
```

Edge Detection



```
[18]: cv2.imwrite('modified_image.jpg', cv2.cvtColor(bright_image, cv2.COLOR_RGB2BGR))
```

```
[18]: True
```