Python programming Lab(23CP301P)

Name: Krishika Vansh

Semester: V

Roll No: 23BCP448

Faculty: Mr. Davinder Singh

Division: VII Batch G13

Branch: Computer Engineering

School of Technology

November 2025

Experiment No: 8

Image Processing using Python Libraries

Objective: To develop a Python program that performs various image processing operations including loading, displaying, manipulating images, and analyzing image histograms.

Code:

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image, ImageFilter, ImageEnhance
import cv2

def load_and_display_image(file_path):

  try:

    img = Image.open(file_path)



    plt.figure(figsize=(8, 6))
    plt.imshow(img)
    plt.title('Original Image')
    plt.axis('off')
    plt.show()
```

```
        return img
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
        return None

def manipulate_image(img):

    if img is None:
        return

    grayscale_img = img.convert('L')
    blurred_img = img.filter(ImageFilter.GaussianBlur(radius=5))



    img_np = np.array(img)

    gray_np = cv2.cvtColor(img_np, cv2.COLOR_RGB2GRAY)

    edges_np = cv2.Canny(gray_np, threshold1=100, threshold2=200)

    edge_img = Image.fromarray(edges_np)



    enhancer = ImageEnhance.Brightness(img)
    bright_img = enhancer.enhance(1.8)
```

```
    fig, axs = plt.subplots(2, 2, figsize=(12, 10))

    axs[0, 0].imshow(grayscale_img, cmap='gray')
    axs[0, 0].set_title('Grayscale')
    axs[0, 0].axis('off')

    axs[0, 1].imshow(blurred_img)
    axs[0, 1].set_title('Gaussian Blur')
    axs[0, 1].axis('off')

    axs[1, 0].imshow(edge_img, cmap='gray')
    axs[1, 0].set_title('Edge Detection (Canny)')
    axs[1, 0].axis('off')

    axs[1, 1].imshow(bright_img)
    axs[1, 1].set_title('Increased Brightness')
    axs[1, 1].axis('off')

    plt.suptitle('Image Manipulations', fontsize=16)
    plt.tight_layout(rect=[0, 0.03, 1, 0.95])
    plt.show()


def analyze_histogram(img):
```

```
    if img is None:
        return
    img_np = np.array(img)

    plt.figure(figsize=(10, 6))
    plt.title('Color Histogram')
    plt.xlabel('Pixel Intensity (0-255)')
    plt.ylabel('Number of Pixels')

    colors = ('r', 'g', 'b')

    for i, color in enumerate(colors):
        # cv2.calcHist([images], [channels], mask, [histSize], [ranges])
        histogram = cv2.calcHist([img_np], [i], None, [256], [0, 256])
        plt.plot(histogram, color=color)
        plt.xlim([0, 256])

    plt.legend(['Red Channel', 'Green Channel', 'Blue Channel'])
    plt.grid(True, linestyle='--', alpha=0.6)
    plt.show()

if __name__ == "__main__":

    image_file = 'myimage.jpeg'
```

```
    original_image = load_and_display_image(image_file)


    manipulate_image(original_image)
    analyze_histogram(original_image)
#install these:
pip install pillow matplotlib numpy opencv-python
```

Output: