Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

COURSE CODE: DJS22ITL603 and Computer Vision Laboratory

COURSE NAME: Image Processing

CLASS: T Y B. TECH

NAME: DIPTI AGARWAL

BATCH : I1-2

SAP: 60003220189

ROLL NO: I047

EXPERIMENT NO. 1

CO/LO: Understand the fundamentals of image processing.

AIM / OBJECTIVE: To explore and understand various Python libraries used for image processing, such as OpenCV, Pillow, and scikit-image.

DESCRIPTION OF EXPERIMENT:

Image processing is a method to perform operations on images to enhance them or extract useful information. Various Python libraries provide powerful tools for image processing, enabling tasks such as image transformation, analysis, and manipulation.

Overview of Libraries:

OpenCV: Open-Source Computer Vision Library, widely used for computer vision tasks. **Pillow:**

A Python Imaging Library (PIL) fork for opening, manipulating, and saving image files. **scikit-**

image: A collection of algorithms for image processing built on top of SciPy.

EXERCISE

1. Install the necessary libraries.
2. Load an image
3. Resize the image
4. Rotate the image

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Information Technology

1. Change the brightness of image
2. Blur the image
3. Visualize the results

CODE:

PILLOW

```python
from PIL import Image

image_path = "image1.jfif"
image = Image.open(image_path)
image
```

```python
print("Original Image Size:", image.size)
```

```
Original Image Size: (225, 225)
```

[27]
```
resized_image = image.resize((300, 300))
resized_image
```

[28]
```
rotated_image = image.rotate(45)
rotated_image
```

[29]

```
from PIL import Image, ImageEnhance, ImageOps, ImageFilter
enhancer = ImageEnhance.Brightness(image)
bright_image = enhancer.enhance(1.5)
bright_image
```

```
blurred_image = image.filter(ImageFilter.GaussianBlur(5))
blurred_image
```

```
grayscale_image = image.convert("L")
grayscale_image
```

```
[32] negative_image = ImageOps.invert(image.convert("RGB"))
     negative_image
```

[33]
```
edge_detected_image = image.filter(ImageFilter.FIND_EDGES)
edge_detected_image
```

[34]
```
embossed_image = image.filter(ImageFilter.EMBOSS)
embossed_image
```

[35]
```
contour_image = image.filter(ImageFilter.CONTOUR)
contour_image

```

[36]
```
sharpened_image = image.filter(ImageFilter.SHARPEN)
sharpened_image

```

A

[37]
```
autocontrast_image = ImageOps.autocontrast(image)
autocontrast_image

```

```
solarized_image = ImageOps.solarize(image, threshold=128)
solarized_image

```

[39]
```
posterized_image = ImageOps.posterize(image, bits=4)
posterized_image

```

[40]
```
equalized_image = ImageOps.equalize(image)
equalized_image

```

```
print("Mode:", image.mode)
print("Format:", image.format)
print("Size:", image.size)
print("Histogram:", image.histogram()[:10])

```

```
Mode: RGB
Format: JPEG
Size: (225, 225)
Histogram: [75, 72, 5, 8, 9, 8, 11, 6, 4, 6]
```

IMAGEIO

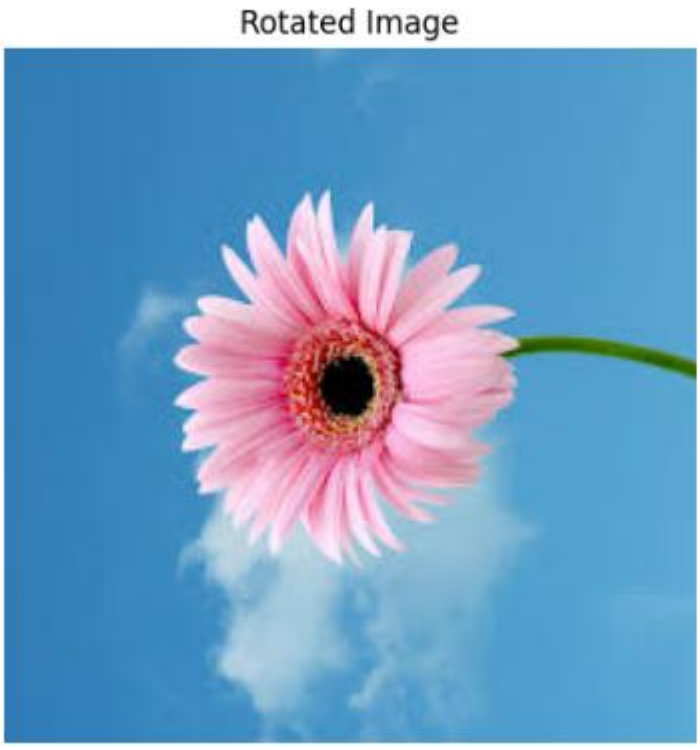
```python
import imageio.v3 as iio
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_ubyte, color, filters, exposure, transfo

def show_image(img, title):
    plt.figure(figsize=(5,5))
    plt.imshow(img, cmap="gray" if len(img.shape) == 2 else None)
    plt.title(title)
    plt.axis("off")
    plt.show()
```

[43]

```python
image_path = "image1.jfif"
image = iio.imread(image_path)
show_image(image, "Original Image")
```

```
resized_image = transform.resize(image, (300, 300), anti_aliasing=T
show_image(resized_image, "Resized Image")
```

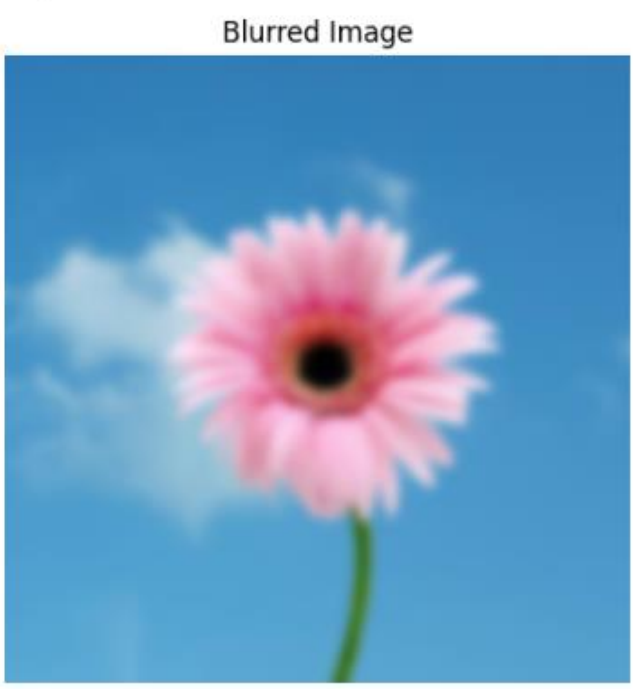

```
rotated_image = np.rot90(image)
show_image(rotated_image, "Rotated Image")
```



[46]
```
grayscale_image = color.rgb2gray(image)
show_image(grayscale_image, "Grayscale Image")
```


```
negative_image = img_as_ubyte(1 - grayscale_image)
show_image(negative_image, "Negative Image")
```

```
1 blurred_image = filters.gaussian(image, sigma=2, channel_axis=-1)
2 show_image(blurred_image, "Blurred Image")
3
```



```
bright_image = exposure.adjust_gamma(image, gamma=0.5)
show_image(bright_image, "Brightened Image")
```

SCKIT-IMAGE

SCKIT-IMAGE

[50]
```python
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color, filters, exposure, transform, util

def show_image(img, title):
    plt.figure(figsize=(5, 5))
    plt.imshow(img, cmap="gray" if len(img.shape) == 2 else None)
    plt.title(title)
    plt.axis("off")
    plt.show()
```

[52]
```python
image_path = "image1.jfif"
image = io.imread(image_path)
show_image(image, "Original Image")
```

```
resized_image = transform.resize(image, (300, 300), anti_aliasing=True)
show_image(resized_image, "Resized Image")
```


```
rotated_image = transform.rotate(image, angle=45)
show_image(rotated_image, "Rotated Image")

```

```
[55] 1 grayscale_image = color.rgb2gray(image)
     2 show_image(grayscale_image, "Grayscale Image")
```


```
1 negative_image = util.invert(image)
2 show_image(negative_image, "Negative Image")
```

```
blurred_image = filters.gaussian(image, sigma=2, channel_axis=-1)
show_image(blurred_image, "Blurred Image")
```


```
bright_image = exposure.adjust_gamma(image, gamma=0.5)
show_image(bright_image, "Brightened Image")
```

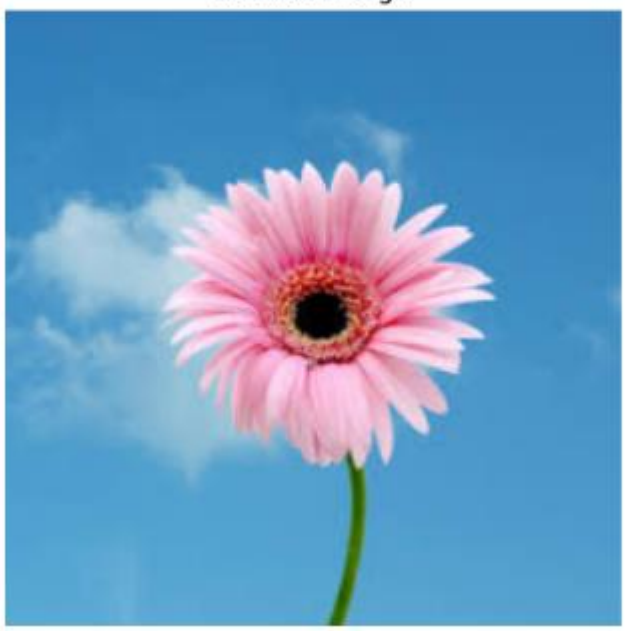
OPEN-CV

Double-click (or enter) to edit

[59]
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def show_image(img, title, cmap=None):
    plt.figure(figsize=(5, 5))
    plt.imshow(img, cmap=cmap)
    plt.title(title)
    plt.axis("off")
    plt.show()
```

[60]
```
image_path = "image1.jfif"
image = cv2.imread(image_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
show_image(image, "Original Image")
```

[61]
```
resized_image = cv2.resize(image, (300, 300))
show_image(resized_image, "Resized Image")
```



[62]
```
(h, w) = image.shape[:2]
center = (w // 2, h // 2)
rotation_matrix = cv2.getRotationMatrix2D(center, 90, 1.0)
rotated_image = cv2.warpAffine(image, rotation_matrix, (w, h))
show_image(rotated_image, "Rotated Image")
```

```
grayscale_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
show_image(grayscale_image, "Grayscale Image", cmap="gray")
```

[64]
```
negative_image = cv2.bitwise_not(grayscale_image)
show_image(negative_image, "Negative Image", cmap="gray")
```

[65]
```
blurred_image = cv2.GaussianBlur(image, (15, 15), 0)
show_image(blurred_image, "Blurred Image")
```

[67]
```
R, G, B = cv2.split(image)
show_image(R, "Red Plane", cmap="Reds")
show_image(G, "Green Plane", cmap="Greens")
show_image(B, "Blue Plane", cmap="Blues")
```

REFERENCE:

Website References:

[1] https://pillow.readthedocs.io/en/stable/

[2] https://pillow.readthedocs.io/en/stable/ [3] https://scikit-image.org/docs/stable/