Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Machine Problem No. 2			
Topic:	**Topic 1.2: Image Processing Techniques**	Week No.:	**3–5**
Course Code:	**CSST106**	Term:	**1st Sem.**
Course Title:	**Perception and Computer Vision**	Academic Year:	**2024-2025**
Student Name:	**Lesly-Ann B. Victoria**	Section:	**BSCS-4B**
Due Date:	**September 21, 2024**	Points:	

Machine Problem No. 2: Applying Image Processing Techniques

Objective: Understand and apply various image processing techniques, including image transformations and filtering, using tools like OpenCV. Gain hands-on experience in implementing these techniques and solving common image processing tasks.

DOCUMENTATION

Installing OpenCV:

```
!pip install opencv-python-headless
```

Result:

```
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.10/dist-packages (4.10.0.84)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python-headless) (1.26.4)
```

Importing Libraries:

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

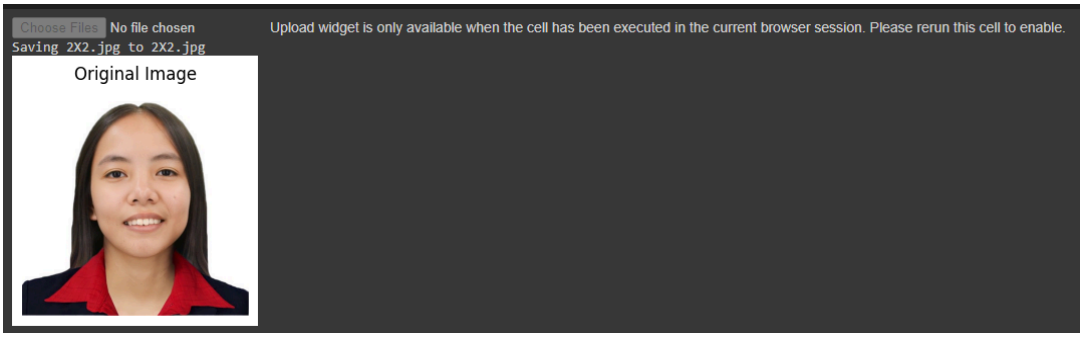
Displaying Image:

```
def display_image(img,title="Image"):
    plt.subplot(1,2,1)
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.title(title)
    plt.axis("off")
    plt.show()

def display_image_gray(img1,img2, title1="Image 1",  title2="Image 2"):
    plt.subplot(1,2,1)
    plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
    plt.title(title1)
    plt.axis("off")


    plt.subplot(1,2,2)
    plt.imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
    plt.title(title2)
    plt.axis("off")

    plt.show()
```

Uploading Image:

```
from google.colab import files
from io import BytesIO
from PIL import Image

uploaded = files.upload()

image_path = next(iter(uploaded))
image = Image.open(BytesIO(uploaded[image_path]))
image = cv2.cvtColor(np.array(image), cv2.COLOR_RGB2BGR)

display_image(image, "Original Image")
```

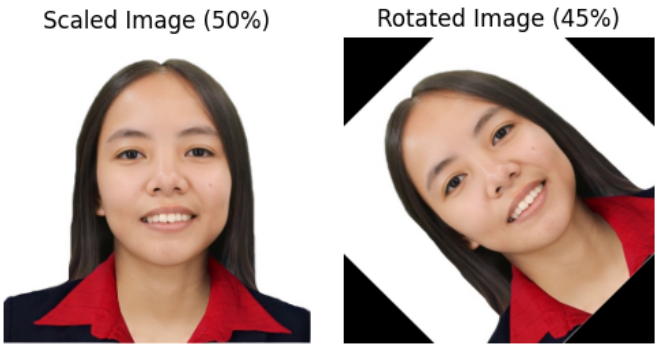
Result:



Exercise 1: Image Transformations (Scaling and Rotation)

```
def scale_image(img, scale_factor):
  height, width = image.shape[:2]
  scale_img = cv2.resize(image, (int(width*scale_factor), int(height*scale_factor)), interpolation=cv2.INTER_LINEAR)
  return scale_img

def rotate_image(image, angle):
  height, width = image.shape[:2]
  center = (width/2, height/2)
  matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
  rotated_img = cv2.warpAffine(image, matrix, (width, height))
  return rotated_img

scaled_image = scale_image(image, 0.5)
display_image(scaled_image, "Scaled Image (50%)")

rotates_image = rotate_image(image, 45)
display_image(rotates_image, "Rotated Image (45%)")
```

Result:



Exercise 2: Filtering Techniques (Blurring Technique)

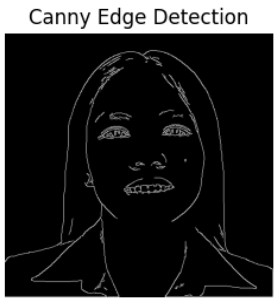
```
gaussian_blur = cv2.GaussianBlur(image, (41, 41), 0)
display_image(gaussian_blur, "Gaussian Blur")

median_blur = cv2.medianBlur(image, 31)
display_image(median_blur, "Median Blur")

bilateral_filter = cv2.bilateralFilter(image, 5, 75, 75)
display_image(bilateral_filter, "Bilateral Filter")
```

Result:

(Edge Detection Technique)

```
edge = cv2.Canny(image, 100, 200)
display_image(edge, "Canny Edge Detection")
```

Result:



In this activity, I explored key image processing techniques using OpenCV, a powerful library for computer vision. I installed OpenCV and successfully loaded and displayed an image, confirming that the library was working correctly. I then applied image transformations such as scaling and rotation, allowing me to resize and reorient images as needed for various tasks. These transformations are essential for adjusting image properties, which is useful in real-world applications like augmented reality and object detection.

Additionally, I applied filtering techniques, including blurring and edge detection. Blurring was used to reduce noise in the image, while edge detection helped highlight object boundaries, which is crucial for identifying shapes and objects. Overall, this project provided hands-on experience with fundamental image processing techniques, enhancing my understanding of how these methods can be applied in AI and computer vision projects.