Machine Problem No. 2			
Topic:	**Topic 1.2: Image Processing Techniques**	**Week No.**	3-5
Course Code:	**CSST106**	**Term:**	1st Semester
Course Title:	**Perception and Computer Vision**	**Academic Year:**	2024-2025
Student Name	**Gajitos, Jude B.**	**Section**	**4B**
Due date	**September 21, 2024**	**Points**	

Machine Problem No. 2: Applying Image Processing Techniques

Step 1: Install OpenCV

```
[13] !pip install opencv-python-headless
```

```
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)
```

Step 2: Import Libraries and Create a Function

```
[14] import cv2
     import numpy as np
     from matplotlib import pyplot as plt

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

Step 3: Upload and Load Image

```
[28] 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")
```

Original Image

Step 4: Image Processing Techniques (Scaling and Rotation)

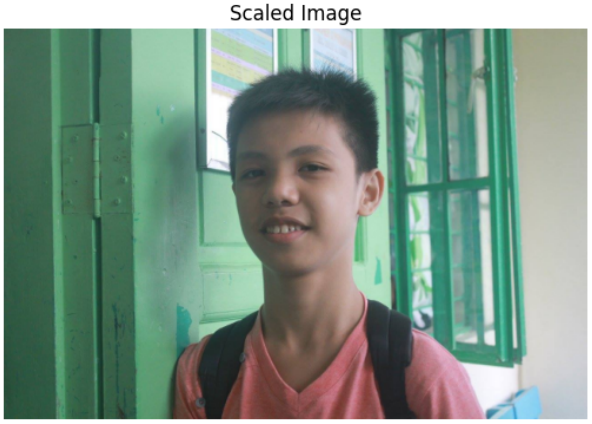
```
[29] def scale_image(image, scale_factor):
       height, width = image.shape[:2]
       scaled_img = cv2.resize(image, (int(width * scale_factor), int(height * scale_factor)), interpolation = cv2.INTER_LINEAR)
       return scaled_img

     def rotate_image(image, angle):
       height, width = image.shape[:2]
       center = (width//2, height//2)
       matrix = cv2.getRotationMatrix2D(center, angle, 1)
       rotated_image = cv2.warpAffine(image, matrix, (width, height))
       return rotated_image

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

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

Scaled Image



Rotated Image

Step 4: Image Processing Techniques (Blurring Techniques)

```
[34] gaussian_blur = cv2.GaussianBlur(image, (23, 23), 0)
     display_image(gaussian_blur, "Gaussian Blur")

     median_blur = cv2.medianBlur(image, 15)
     display_image(median_blur, "Median Blur")

     bilateral_blur = cv2.bilateralFilter(image, 105, 75, 75)
     display_image(bilateral_blur, "Bilateral Blur")
```

Gaussian Blur

Median Blur

Bilateral Blur

Step 4: Image Processing Techniques (Edge Detection using Canny)

```
[43] edge_detection = cv2.Canny(image, 45, 95)
     display_image(edge_detection, "Edge Detection")
```

Edge Detection