Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

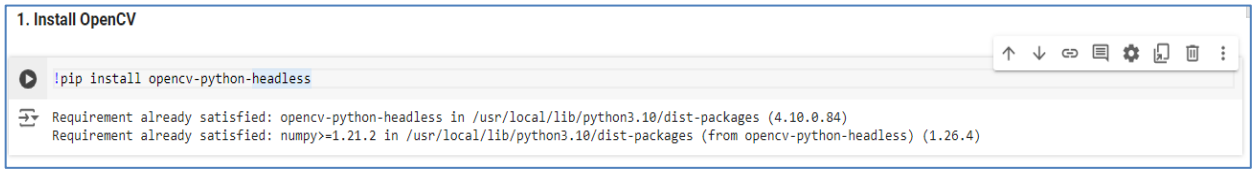
Machine Problem No. 2			
Topic:	**Topic 1.2: Image Processing Techniques**	**Week No.**	3-5
Course Code:	**CSST106**	**Term:**	1st Semester
Course Title:	**Perception and Computer Vision**	**Academic Year:**	2024-2025
Student Name	**Simon B. Sancon**	**Section**	
Due date	**September 21, 2024**	**Points**	

Machine Problem No. 2: Applying Image Processing Techniques

Report

This Report include the steps taken and the result of the first exercise image processing techniques

Step 1: Installing OpencV



Step 2: Import Libraries

2. Import Libraries

```
[2]  import cv2
     import numpy as np
     import matplotlib.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()

     def display_images(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()
```

Step 3: Load 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")
```

```
Choose Files  image1 (1).jpg
  image1 (1).jpg(image/jpeg) - 54683 bytes, last modified: 9/9/2024 - 100% done
Saving image1 (1).jpg to image1 (1).jpg
```

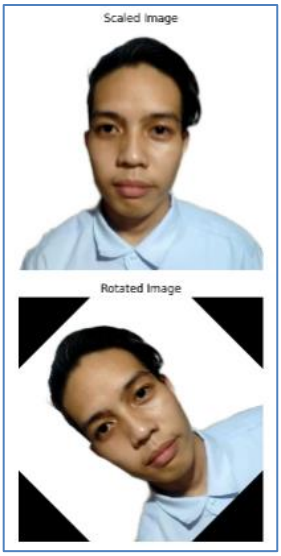
4. Exercise 1: Scaling and Orientation

```
def scale_image(image, 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)
  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")
```



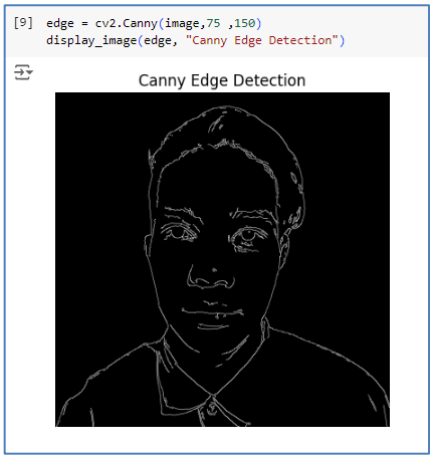
5. Blurring Image

```
guassian_blur = cv2.GaussianBlur(image,(61,61),0)
display_image(guassian_blur,"Guassian Blur")

median_blur = cv2.medianBlur(image,11)
display_image(median_blur,"Median Blur")

bilateral_blur = cv2.bilateralFilter(image,99,75,75)
display_image(bilateral_blur,"Bilateral Blur")
```

Step 6: Edge Detection Using Canny

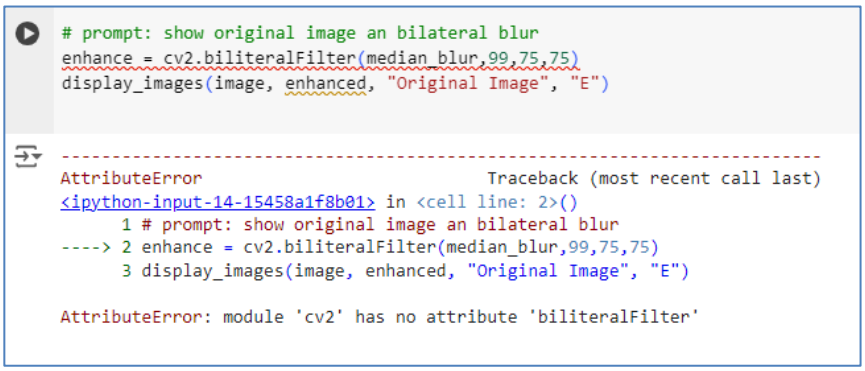


Problem-Solving Session

Common Challenges

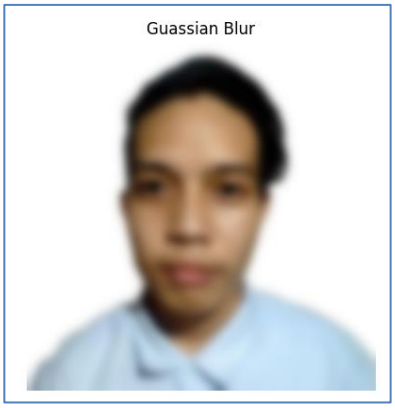
There are 2 common challenges that was experienced during the Problem-Solving Session

1. Syntax error



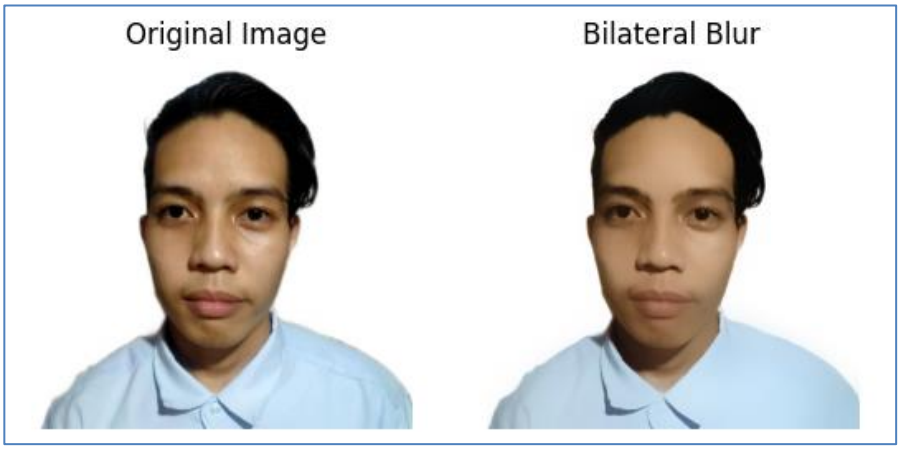
The figure above is one common issue that will be tackled whenever an implementation was conducted, this includes spelling, misused variables, and unfamiliarity of syntax or library

2. Finding the right value



The figure above shows that finding the right value of the applied image processing techniques requires some trial and error to produce the desired result.

Scenario-Based Problems:



The figure above shows that we can apply Blurring to enhance a photo to reduce noise from the photo