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	**Lucky Owell U. Millare**	**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.

Hands-On Exploration:

1. INSTALL OPENCV

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

2. IMPORT LIBRARIES

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

def display_image(img, title="Image"):
  plt.figure(figsize=(6,3))
  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.figure(figsize=(6,3))
  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()
```

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")
```

Original Image

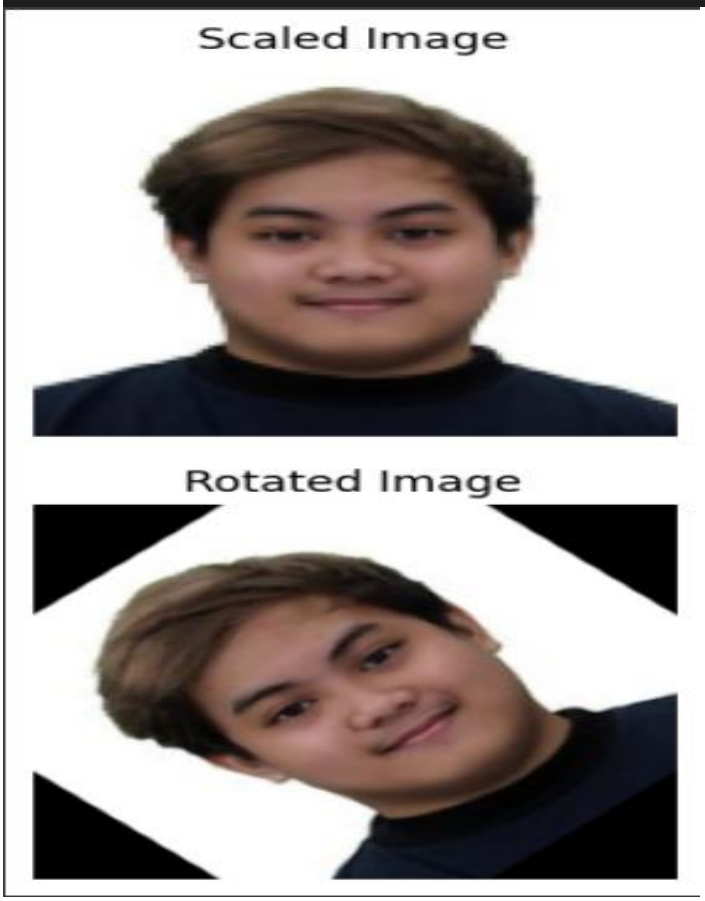
EXERCISE 1. Scaling and Rotation

```
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")
```



Exercise 2: Blurring Techniques

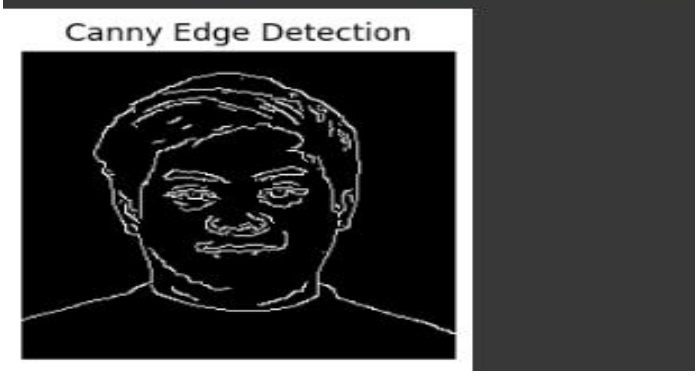
```
gussian_blur = cv2.GaussianBlur(image, (11,11), 0)
display_image(gussian_blur, "Gussian Blur")

median_blur = cv2.medianBlur(image,17)
display_image(median_blur, "Median Blur")

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

3. Edge Detection using Canny

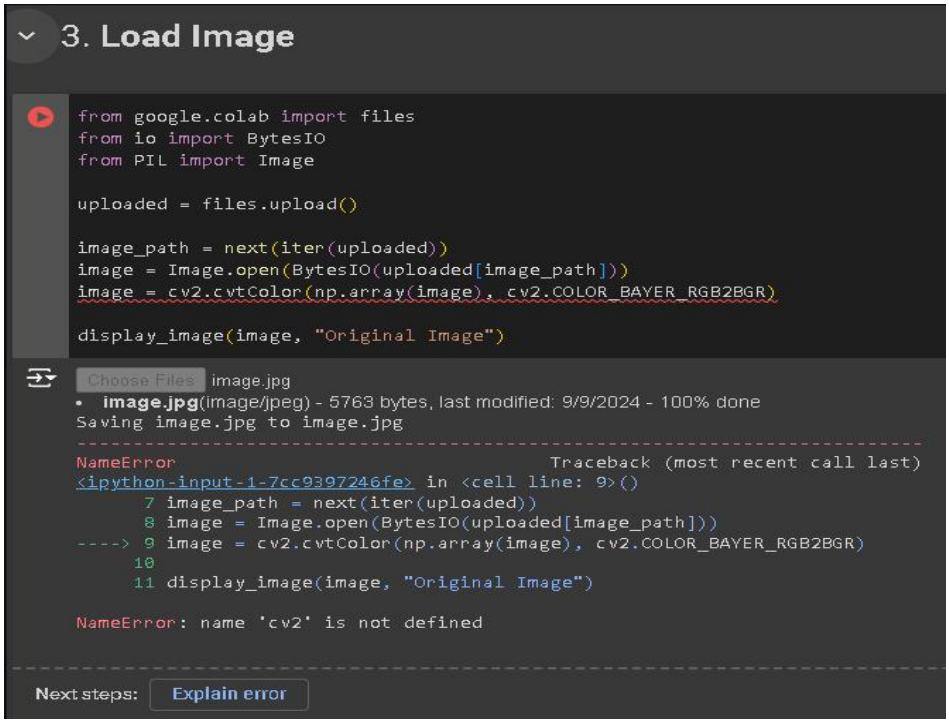
```
edge = cv2.Canny(image,100 ,150)
display_image(edge, "Canny Edge Detection")
```



Problem-Solving Session:

- **Common Image Processing Tasks:**
  - Engage in a problem-solving session focused on common challenges encountered in image processing tasks.
  -

One of the Challenge I encounter in our lab session is the uploading a image in colab notebook this is because of having a BAYER in the code because of TAB too much in the Laboratory session.



To fix the problem is we have to remove the BAYER in the code so we can upload a image without error.

Scenario-Based Problems: Solve scenarios where you must choose and apply appropriate image processing techniques.

In the Exercise 1 Hand on Exploration we following the step of our professor instructed in the TV screen the problem I encounter in the one of the Image processing Techniques that been instructing is in the Scaling and Rotation. The problem I encounter in that Part is I didn't copy properly the code the professor instructed because I TAB it without knowing the code is not the same so when I run the code there is a error.

```
def scale_image(image, scale_factor):
  height, width = image.shape[:2]
  scale_img = cv2.resize(image, 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")
```

```
---------------------------------------------------------------------------
error                                     Traceback (most recent call last)
<ipython-input-17-b0f5ac9ca581> in <cell line: 13>()
     11   return rotated_image
     12
---> 13 scaled_image = scale_image(image, 0.5)
     14 display_image(scaled_image,"Scaled Image")
     15

<ipython-input-17-b0f5ac9ca581> in scale_image(image, scale_factor)
      1 def scale_image(image, scale_factor):
      2   height, width = image.shape[:2]
----> 3   scale_img = cv2.resize(image, int(height * scale_factor), interpolation = cv2.INTER_LINEAR)
      4   return scale_img
      5

error: OpenCV(4.10.0) :-1: error: (-5:Bad argument) in function 'resize'
> Overload resolution failed:
>  - Can't parse 'dsize'. Input argument doesn't provide sequence protocol
>  - Can't parse 'dsize'. Input argument doesn't provide sequence protocol
```

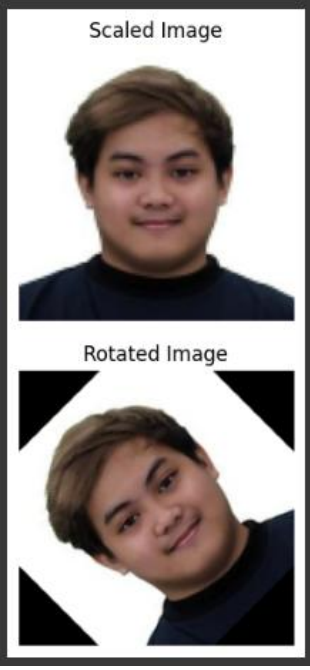
To solve the problem I try to compare my code to my seatmates and learn what is wrong in my code the problem is in the scale_img I do not have this (int(width * scale_factor) that will be used in to calculate the new width of the image after scaling. After I fix it the error is gone.

```python
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")
```



Rubric for Machine Problem No. 2: Applying Image Processing Techniques

Criteria	Excellent (10 points)	Good (8 points)	Fair (5 points)	Poor (2 points)
Understanding of Image Processing Concepts	Comprehensive understanding of image processing techniques.	Good understanding with minor gaps.	Basic understanding with some inaccuracies.	Poor or incomplete understanding.
Application of Transformations and Filters	Correct and effective application of image transformations and filters.	Mostly correct with minor inaccuracies.	Basic application with significant errors.	Inaccurate or ineffective application.
Problem-Solving Ability	Effective solutions to common image processing tasks.	Adequate solutions with some errors.	Limited solutions with many errors.	Poor or incorrect solutions.
Lab Work Submission	Well-organized, clear, and complete submission.	Organized with minor issues in clarity.	Somewhat organized but lacks clarity.	Disorganized and unclear submission.