# Machine Problem No. 2: Applying Image Processing Techniques

### Step 1: Install OpenCV

!pip install opency-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 Necessary Libraries

```
import cv2
import numpy as np
import matipatith.pyplot as plt
from google.colab import files
from in import BytesIO
from Plt import lange

# Function to display an image using matplotlib
def display_image(img, title="Image"):
    plt.imshow(cv2.cvtcolor(img, cv2.ColoR_BGR2RGB))
    plt.title(title)
    plt.axis('off')
    plt.noshow()

# Function to display two images side by side
def display_images(img1, img2, title1="Image 1", title2="Image 2"):
    plt.imshow(cv2.cvtcolor(img1, cv2.ColoR_BGR2RGB))
    plt.timshow(cv2.cvtcolor(img1, cv2.ColoR_BGR2RGB))
    plt.imshow(cv2.cvtcolor(img2, cv2.ColoR_BGR2RGB))
    plt.swbplot(1, 2, 2)
    plt.imshow(cv2.cvtcolor(img2, cv2.ColoR_BGR2RGB))
    plt.title(title2)
    plt.smbplot(1, 2, 2)
    plt.imshow(cv2.cvtcolor(img2, cv2.ColoR_BGR2RGB))
    plt.title(title2)
    plt.axis('off')

plt.show()
```

#### Step 3: Load an Image

```
# Upload an image
uploaded = files.upload()

# Convert to OpenCV format
image_path = next(iter(uploaded)) # Get the image file name
image = Image.open(BytesIO(uploaded[image_path]))
image = cv2.cvtColor(np.array(image), cv2.ColOR_RGB2BGR)

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

Choose Files cat.jpg

• cat.jpg(image/peg) - 40670 bytes, last modified: 9/19/2024 - 100% done Saving cat.jpg

Original Image

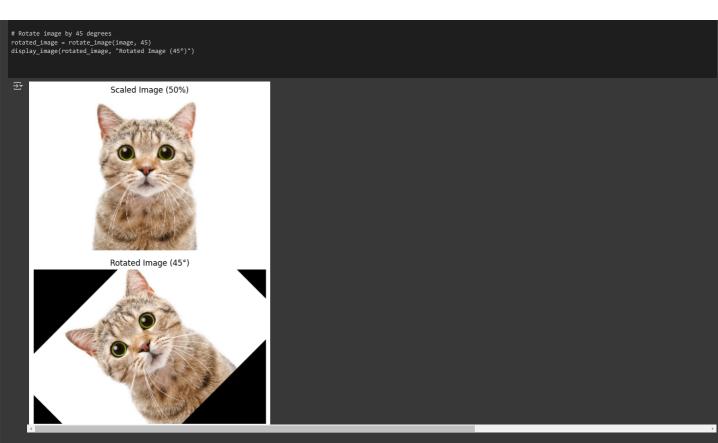
Implementing Image Transformations and Filtering:

#### Scaling and Rotation

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

# Rotate
def rotate_image(img, angle):
    height, width = img.shape[:2]
    center = (width // 2, height // 2)
    matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
    rotated_img = cv2.varpAffine(img, matrix, (width, height))
    return rotated_img

# Scale_image by 0.5
scaled_image = scale_image(image, 0.5)
display_image(scaled_image, "Scaled Image (50%)")
```



## Blurring Techniques

# Gaussian Blur
gaussian\_blur = cv2.GaussianBlur(image, (5, 5), 0)
display\_image(gaussian\_blur, "Gaussian Blur (5x5)")
# Median Blur

# Median Blur
median\_blur = cv2.medianBlur(image, 5)
display\_image(median\_blur, "Median Blur (5x5)")





## Edge Detection using Canny

# Canny Edge Detection
edges = cv2.Canny(image, 100, 200)
display\_image(edges, "Canny Edge Detection (100, 200)")