

**Computer Vision**

**MCA-574**

**Assignment – 01**

***BY***

**HIMANSHU HEDA (24225013)**

**SUBMITTED TO**

**Dr. Preety Shoran**

**SCHOOL OF SCIENCES**

**2024-25**

**# Computer Vision OpenCV Tasks**

This notebook demonstrates the following tasks using OpenCV:

1. Read and display an image in color and grayscale.

2. Resize and rotate an image.

3. Convert an image from RGB to HSV and Grayscale.

4. Draw basic shapes (line, rectangle, circle) and add text on an image.

5. Flip (horizontal/vertical) and crop a region of interest (ROI) from an image.

**Code : --**

# 1. Read and display an image in color and grayscale

import cv2

from matplotlib import pyplot as plt

# Read the image in color

image = cv2.imread('pika wallpaper.webp')

if image is None:

    print("Error: Could not open or find the image.")

else:

    # Convert to grayscale

    gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    # Display using matplotlib for notebook compatibility

    plt.figure(figsize=(10,4))

    plt.subplot(1,2,1)

    plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

    plt.title('Color Image')

    plt.axis('off')

    plt.subplot(1,2,2)

    plt.imshow(gray\_image, cmap='gray')

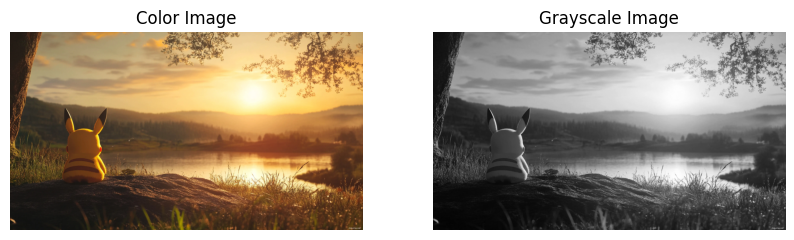
    plt.title('Grayscale Image')

    plt.axis('off')

    plt.show()

    # Save grayscale image

    cv2.imwrite('gray\_pika\_wallpaper.png', gray\_image)

****

# 2. Resize and rotate an image

import numpy as np

# Resize image to 400x400 and rotate by 45 degrees

if image is not None:

    resized = cv2.resize(image, (400, 400))

    # Rotation

    center = (200, 200)

    angle = 45

    scale = 1.0

    M = cv2.getRotationMatrix2D(center, angle, scale)

    rotated = cv2.warpAffine(resized, M, (400, 400))

    # Display

    plt.figure(figsize=(10,4))

    plt.subplot(1,2,1)

    plt.imshow(cv2.cvtColor(resized, cv2.COLOR\_BGR2RGB))

    plt.title('Resized (400x400)')

    plt.axis('off')

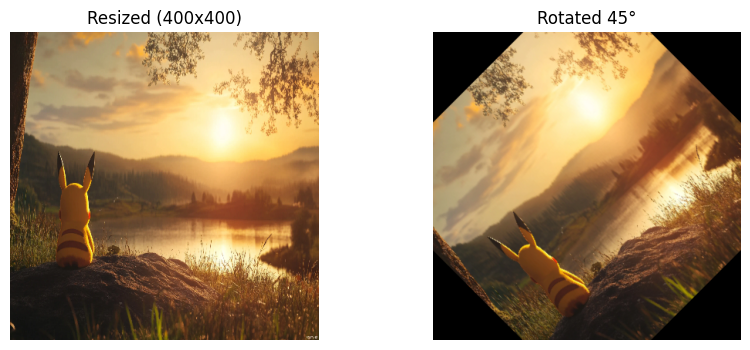
    plt.subplot(1,2,2)

    plt.imshow(cv2.cvtColor(rotated, cv2.COLOR\_BGR2RGB))

    plt.title('Rotated 45°')

    plt.axis('off')

    plt.show()

****

# 3. Convert an image from RGB to HSV and Grayscale

if image is not None:

    hsv\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

    gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    plt.figure(figsize=(15,4))

    plt.subplot(1,3,1)

    plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

    plt.title('RGB Image')

    plt.axis('off')

    plt.subplot(1,3,2)

    plt.imshow(hsv\_image[:,:,0], cmap='hsv')

    plt.title('HSV (Hue channel)')

    plt.axis('off')

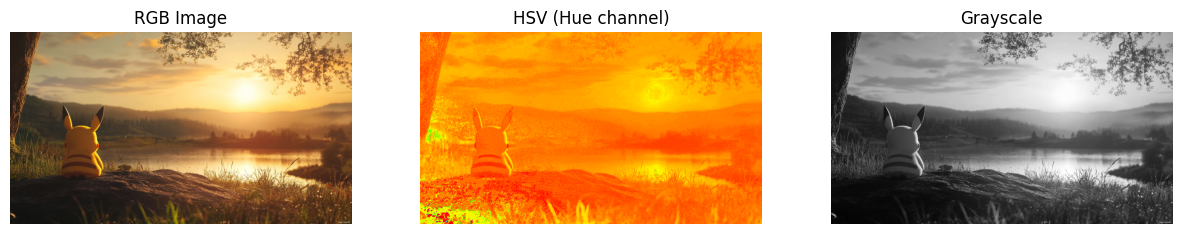
    plt.subplot(1,3,3)

    plt.imshow(gray\_image, cmap='gray')

    plt.title('Grayscale')

    plt.axis('off')

    plt.show()

****

# 4. Draw basic shapes and add text on an image

if image is not None:

    img\_shapes = image.copy()

    # Draw a blue line

    cv2.line(img\_shapes, (50, 50), (350, 50), (255, 0, 0), 5)

    # Draw a green rectangle

    cv2.rectangle(img\_shapes, (50, 100), (350, 200), (0, 255, 0), 3)

    # Draw a red circle

    cv2.circle(img\_shapes, (200, 300), 50, (0, 0, 255), -1)

    # Add text

    cv2.putText(img\_shapes, 'OpenCV Demo', (60, 380), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0,0,0), 2, cv2.LINE\_AA)

    # Display

    plt.figure(figsize=(6,6))

    plt.imshow(cv2.cvtColor(img\_shapes, cv2.COLOR\_BGR2RGB))

    plt.title('Shapes and Text')

    plt.axis('off')

    plt.show()

****

# 5. Flip (horizontal/vertical) and crop a region of interest (ROI) from an image

if image is not None:

    # Flip horizontally and vertically

    flip\_h = cv2.flip(image, 1)

    flip\_v = cv2.flip(image, 0)

    # Crop ROI (center 200x200 region)

    h, w = image.shape[:2]

    x, y = w//2 - 100, h//2 - 100

    roi = image[y:y+200, x:x+200]

    # Display

    plt.figure(figsize=(15,4))

    plt.subplot(1,3,1)

    plt.imshow(cv2.cvtColor(flip\_h, cv2.COLOR\_BGR2RGB))

    plt.title('Flipped Horizontal')

    plt.axis('off')

    plt.subplot(1,3,2)

    plt.imshow(cv2.cvtColor(flip\_v, cv2.COLOR\_BGR2RGB))

    plt.title('Flipped Vertical')

    plt.axis('off')

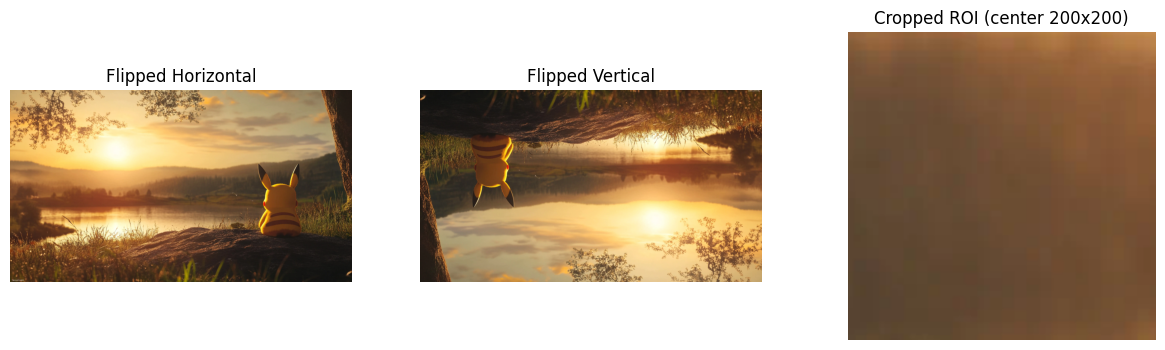
    plt.subplot(1,3,3)

    plt.imshow(cv2.cvtColor(roi, cv2.COLOR\_BGR2RGB))

    plt.title('Cropped ROI (center 200x200)')

    plt.axis('off')

    plt.show()

****