Department of Computer Science and Engineering (Data Science)

Image Processing and Computer Vision I (DJ19DSL603) Lab 1: Arithmetic and Logical Operations on Image

Name: Yash Thakar SAP ID.: 60009210205

Aim: To Perform Basic Image Processing Operations in Python.

Problem Statement: Develop a Python program utilizing the OpenCV library to manipulate images from the MNIST digits dataset. The program should address the following tasks:

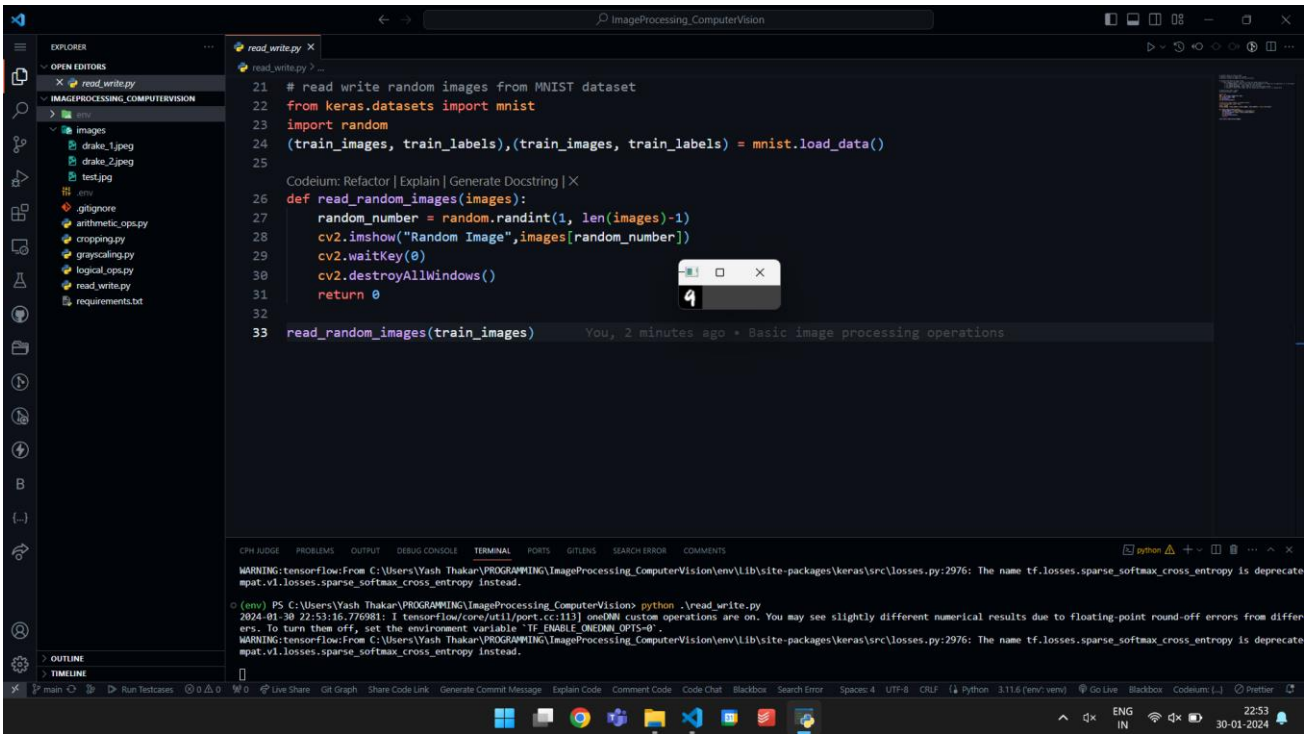
1. Read random image(s) from the MNIST digits dataset given the digit value.
2. Display the before & after image(s) used in every task below.
3. Convert the image to grayscale.
4. Implement image cropping functionality.
5. Perform arithmetic operations on the images - Addition & Subtraction.
6. Implement logical operations on the image(s) - AND, OR, NOT & XOR.

The solution to the operations performed must be produced by scratch coding without the use of built in OpenCV methods.

Dataset Link: MNIST Digits Kaggle

Results:

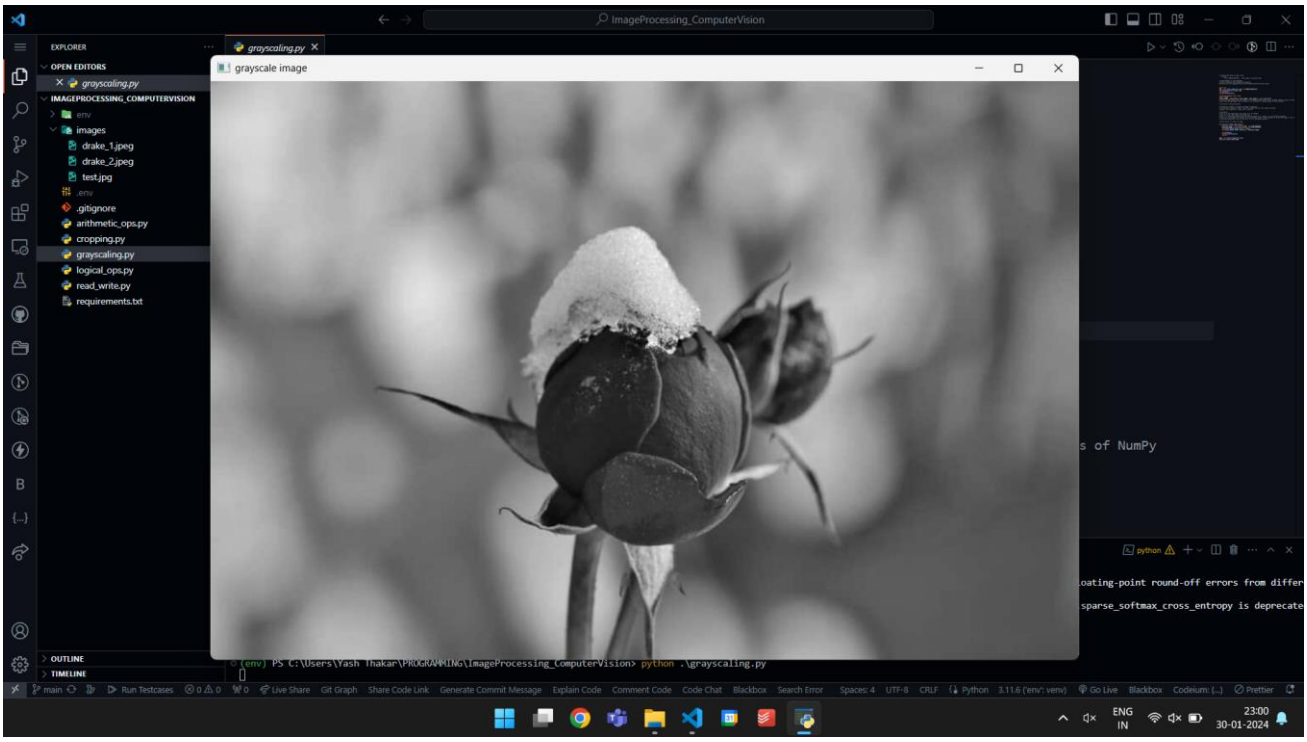
Read random image(s) from the MNIST digits dataset given the digit value.

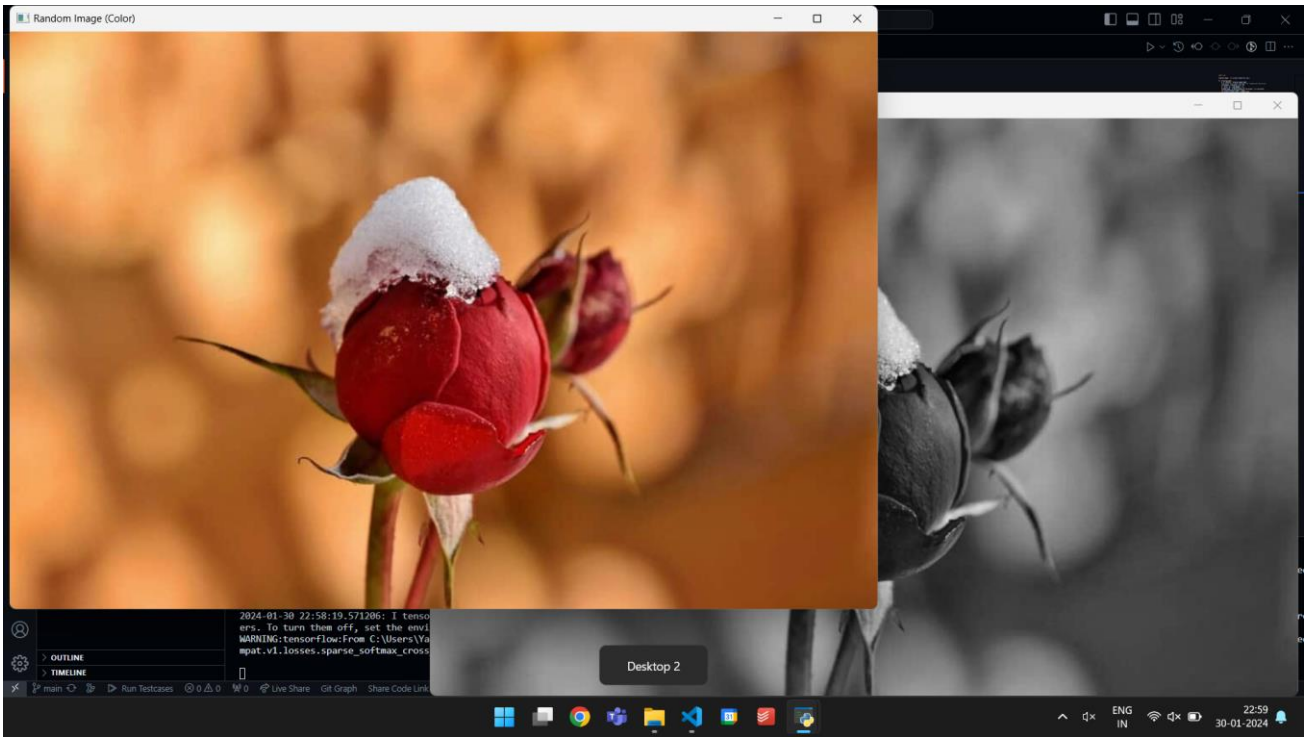


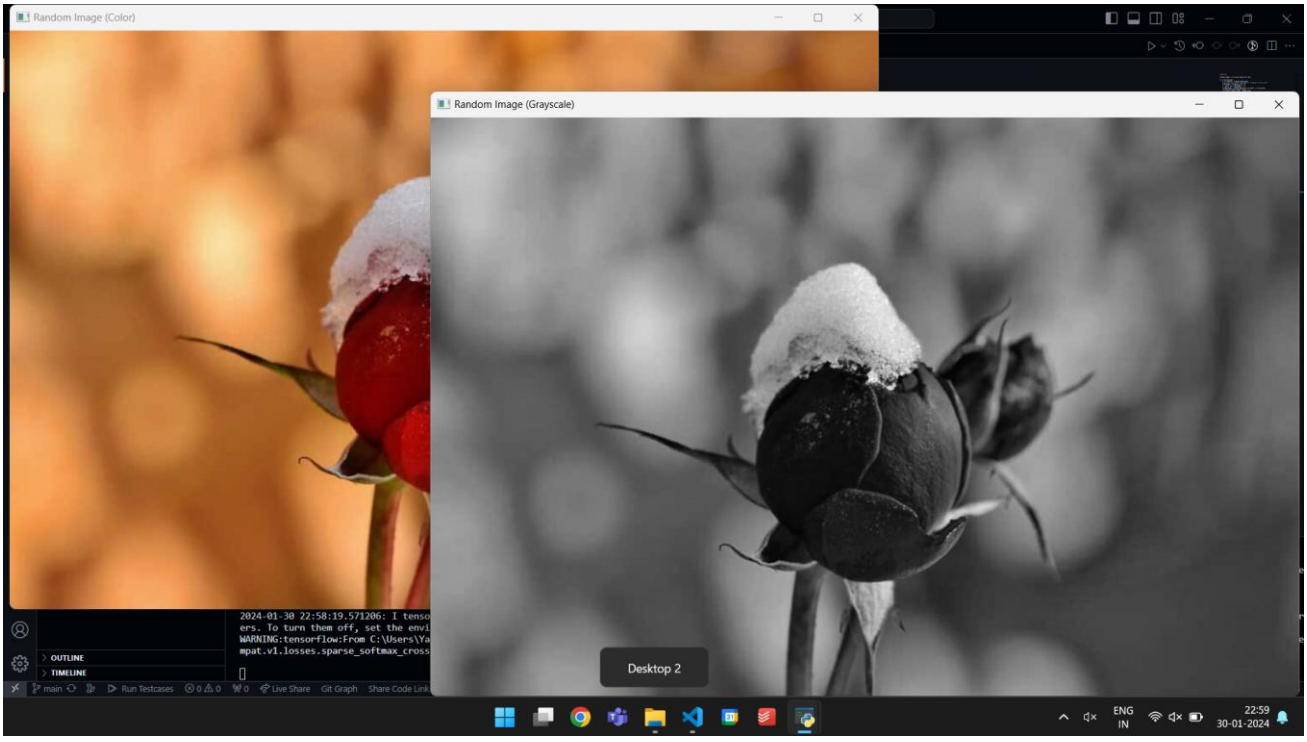
Convert the image to grayscale.

```
You, 6 minutes ago | 1 author (You) | Click here to ask Blackbox to help you code faster
# imread(folder/path_to_image, flag)
    # flag:
        # cv2.IMREAD_GRAYSCALE : Loads image in grayscale mode

# COLOR_BGR2GRAY vs COLOR_BGR2GRAY
# https://stackoverflow.com/questions/62855718/
why-would-cv2-color-rgb2gray-and-cv2-color-bgr2gray-give-different-results


import cv2
img = cv2.imread('images/test.jpg',cv2.IMREAD_GRAYSCALE)
cv2.imshow('grayscale image',img)
cv2.waitKey(0)      You, 5 minutes ago • Basic image processing operations
cv2.destroyAllWindows()

from keras.datasets import mnist
import random
(train_images, train_labels),(train_images, train_labels) = mnist.load_data()
```

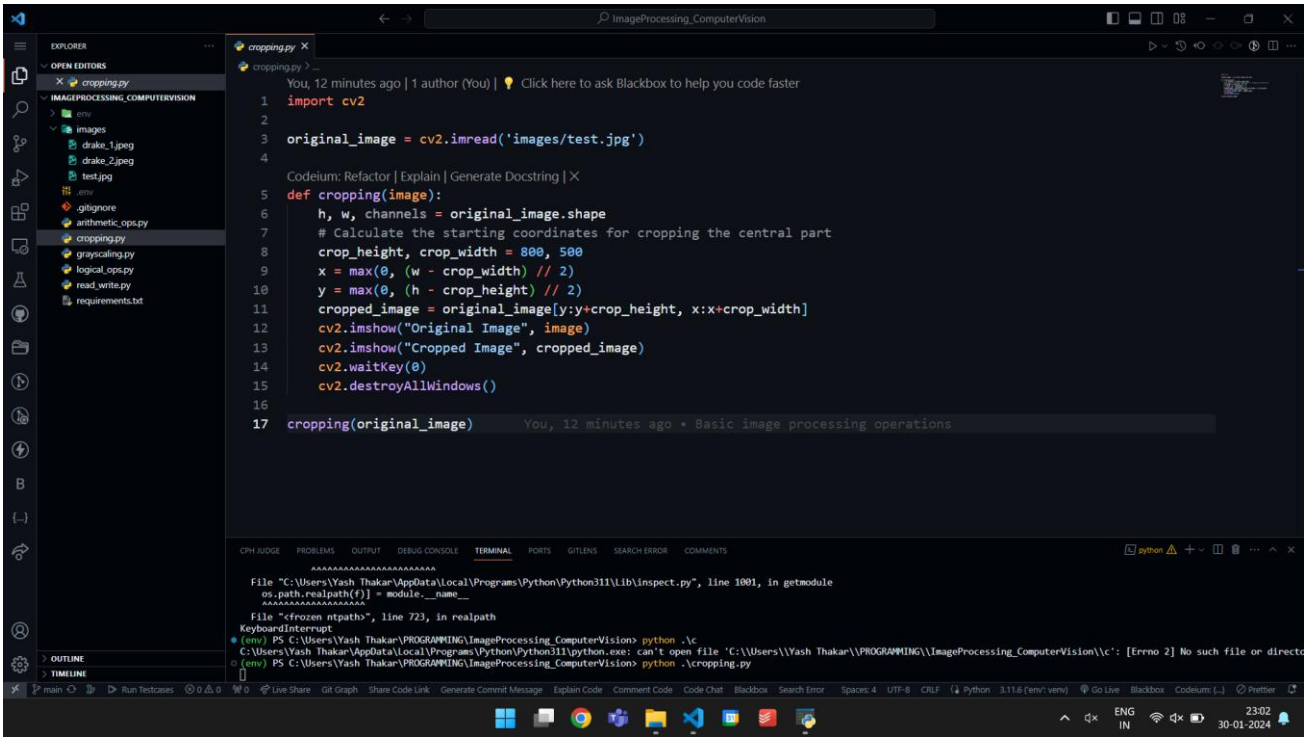
```
Codeium: Refactor | Explain | Generate Docstring | X
def grayscale_random_images(images):
    #grayscale_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    grayscale_image = cv2.cvtColor(images, cv2.COLOR_RGB2GRAY)
    cv2.imshow("Random Image (Color)", images)
    cv2.imshow("Random Image (Grayscale)", grayscale_image)

    cv2.waitKey(0)
    cv2.destroyAllWindows()
    return 0

img2 = cv2.imread('images/test.jpg')
grayscale_random_images(img2)
```

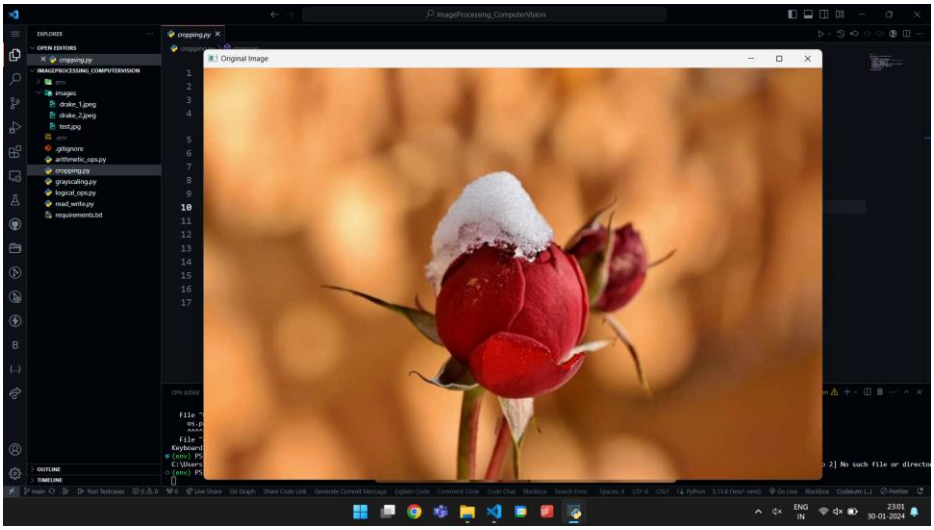


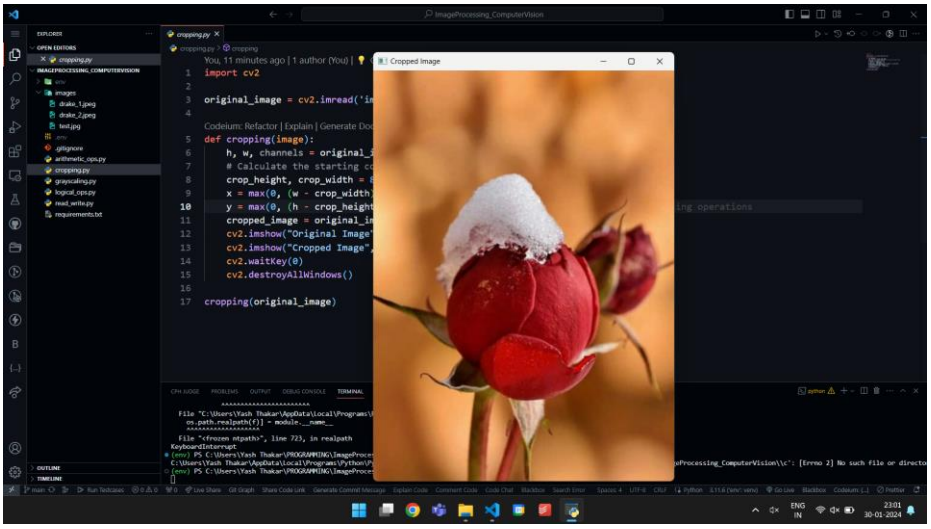

Random Image (Color)
2024-01-30 22:58:19.571206: I tenso
ers. To turn them off, set the envi
WARNING:tensorflow:From C:\Users\Ya
mpat.v1.losses.sparse_softmax_cross
OUTLINE
TIMELINE
Desktop 2
ENG
IN
22:59
30-01-2024


Random Image (Color)
Random Image (Grayscale)
2024-01-30 22:58:19.571206: I tenso
ers. To turn them off, set the envi
WARNING:tensorflow:From C:\Users\Ya
mpat.v1.losses.sparse_softmax_cross
OUTLINE
TIMELINE
Desktop 2
ENG
IN
22:59
30-01-2024

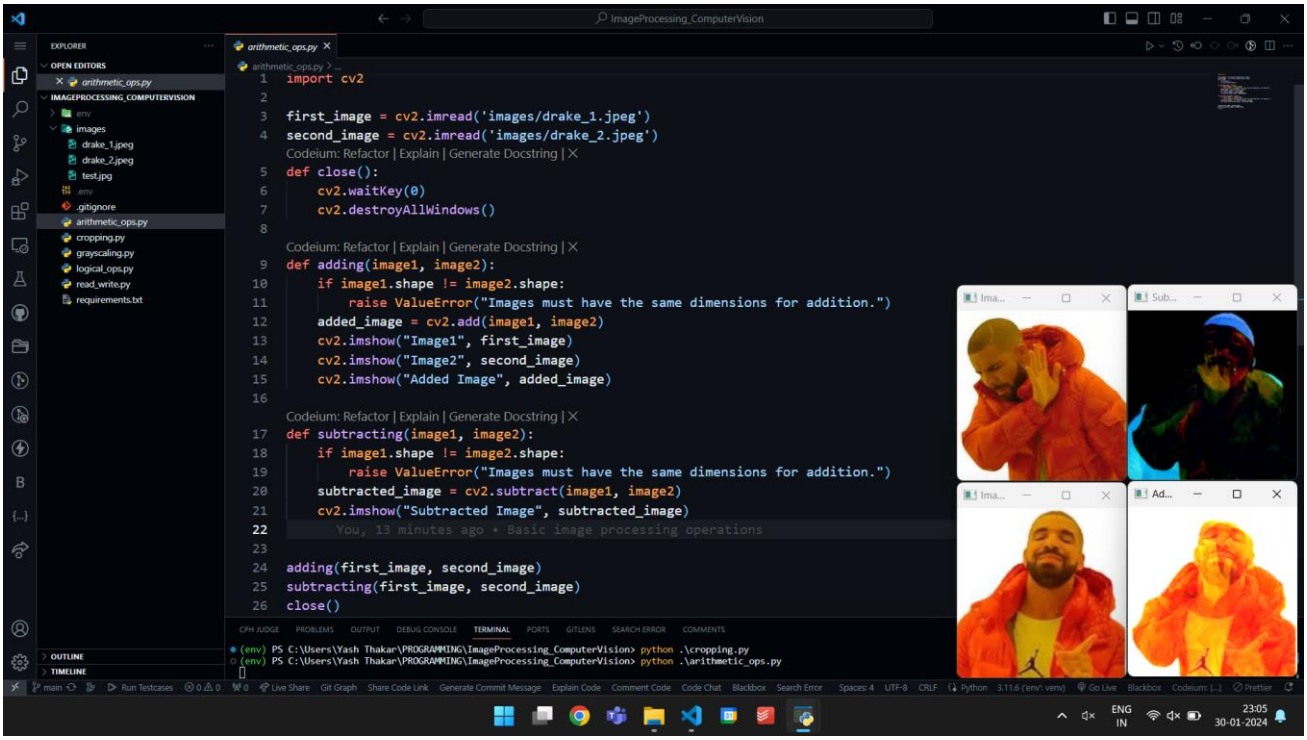
Implement image cropping functionality.







Perform arithmetic operations on the images - Addition & Subtraction.



Implement logical operations on the image(s) - AND, OR, NOT & XOR.

```
You, 12 seconds ago | 1 author (You) | 💡 Click here to ask Blackbox to help you code faster
import cv2

Codeium: Refactor | Explain | Generate Docstring | X
def logical_and(image1, image2):
    img1 = cv2.imread(image1)
    img2 = cv2.imread(image2)

    img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
    result = cv2.bitwise_and(img1, img2)

    return result

Codeium: Refactor | Explain | Generate Docstring | X
def logical_or(image1, image2):
    img1 = cv2.imread(image1)
    img2 = cv2.imread(image2)

    img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
    result = cv2.bitwise_or(img1, img2)

    return result

```

```
Codeium: Refactor | Explain | Generate Docstring | X
def logical_not(image):
    img = cv2.imread(image)

    result = cv2.bitwise_not(img)

    return result

Codeium: Refactor | Explain | Generate Docstring | X
def logical_xor(image1, image2):
    img1 = cv2.imread(image1)
    img2 = cv2.imread(image2)
    img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
        You, 22 minutes ago • Basic image processing operations
    result = cv2.bitwise_xor(img1, img2)

    return result

image1_path = 'images\drake_1.jpeg'
image2_path = 'images\drake_2.jpeg'
```

```python
result_and = logical_and(image1_path, image2_path)
cv2.imshow('AND', result_and)

result_or = logical_or(image1_path, image2_path)
cv2.imshow('OR', result_or)

result_not = logical_not(image1_path)
cv2.imshow('NOT', result_not)

result_xor = logical_xor(image1_path, image2_path)
cv2.imshow('XOR', result_xor)

cv2.waitKey(0)
cv2.destroyAllWindows()
```