PRACTICAL -1

Name: Shivanshu Anant Suryakar

PRN: 1841048

Batch: B3

Class: L.Y Computer Engineering

Aim: Write a program to enhance an image using image arithmetic and logical operations

---

Theory:

There are two different types of operations that are widely used in image processing especially in image morphology.

Arithmetic Operations

Arithmetic operations between two pixels a and b are denoted as follows:

i) Addition: a + b

Image addition is used in image averaging to reduce noise. This kind of operation was performed in image enhancement.

ii) Subtracting: a - b

Image subtraction is widely used in medical imaging. A very common example is the Digital Subtraction Angiography (DSA).

Image Subtraction is basically used to get rid of background information.

iii) Image multiplication: a × b

iv) Image division: a/b

Both, Image Multiplication and Image Division are used to correct grey level shading that result from non-uniformities in illumination or in the sensor used.

One thing that needs to be understood in arithmetic operations is that these are carried out on an entire image, pixel by pixel. Arithmetic operations involve only

one spatial pixel location at a time and hence can be done in place. Arithmetic operations are applied to multi-valued pixels.

Logical Operations

Logical operations commonly used are as follows:

a) AND : a AND b

b) OR : a OR b

c) COMPLEMENT : NOT a

These operations can be combined to form other logic operations. Logic operations actually apply only to binary images. (Image having only two values, 0 and 1).

The AND operator gives out 1 only when both 'a' and 'b' are equal to 1.
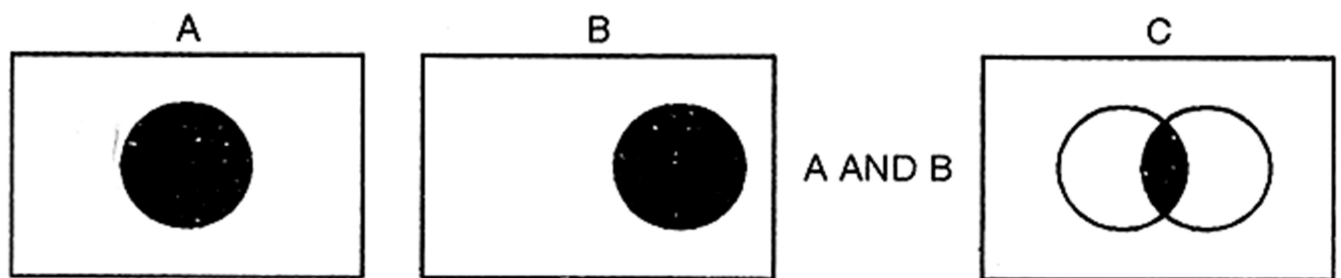
Figure 37

The OR operator gives out 1 if either 'a' or 'b' or both are equal to 1.
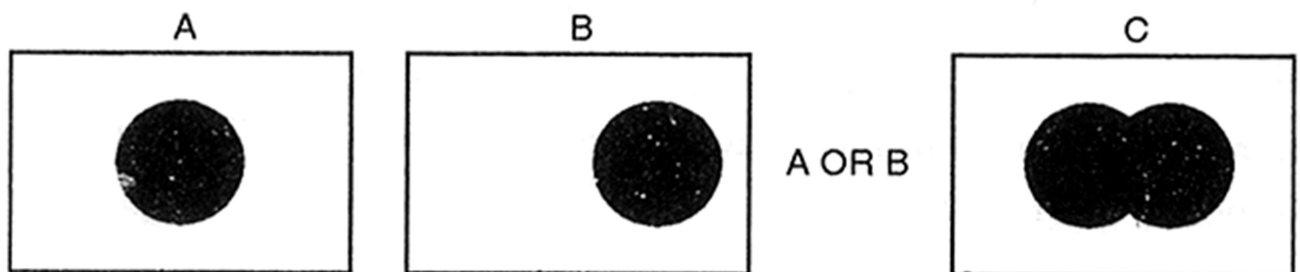
The COMPLEMENT (NOT) operator gives out 1 when a = 0

A NOT (A)

The AND operator applied on sets A and B is also denoted in the jargon of set theory as

C = A ∩ B
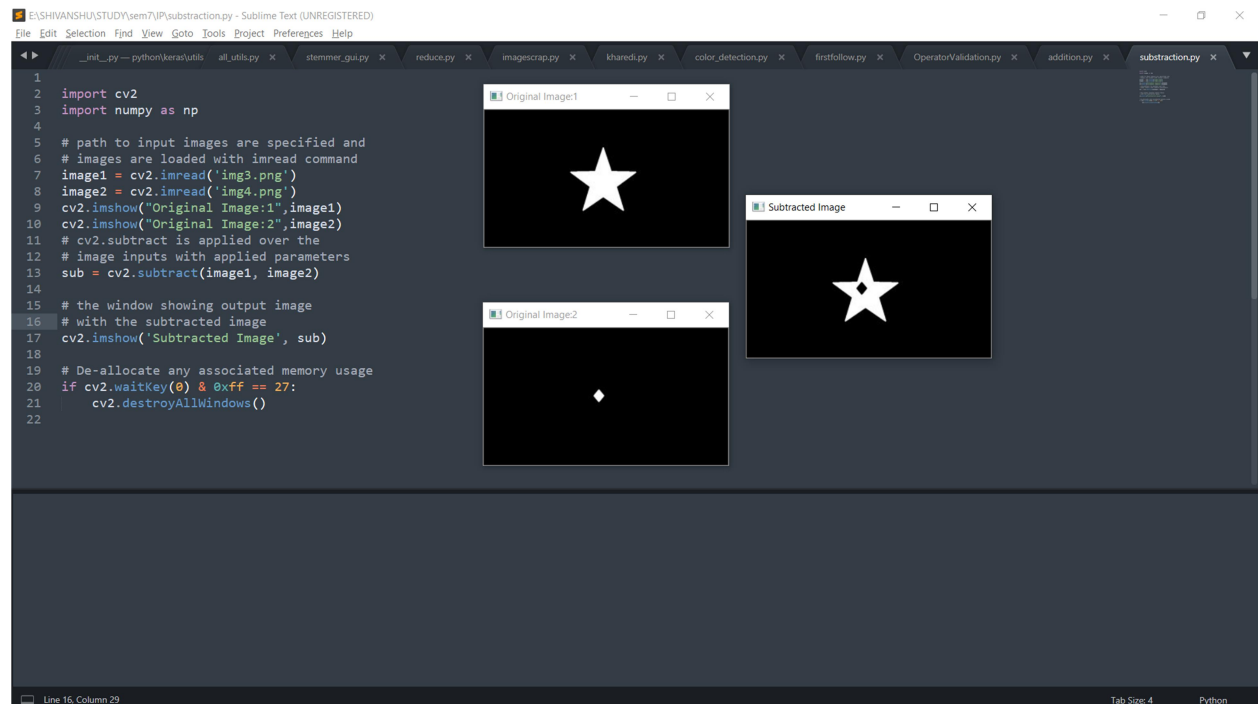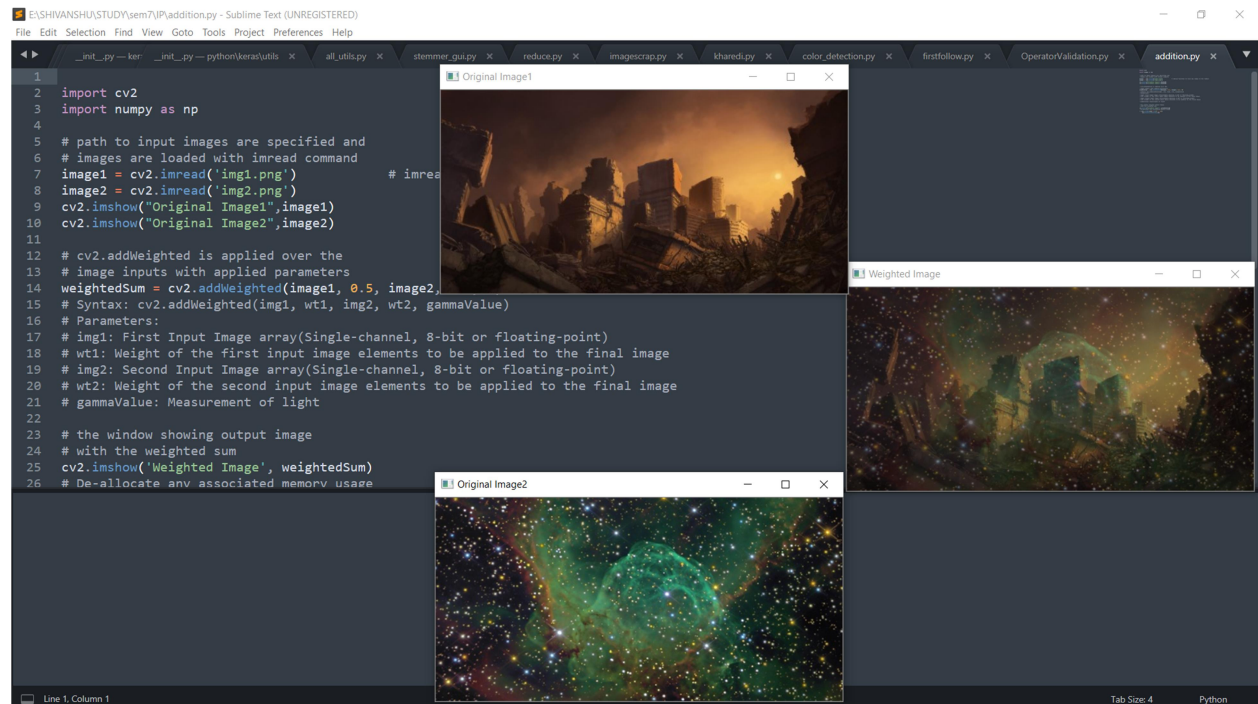i.e., C is the intersection of A and B. Similarly, the OR operator applied on sets A and B is denoted as

C = A ∪ B
i.e., C is the union of A and B.

Requirements:

- Open CV
- Numpy

Output:


```python
import cv2
import numpy as np

# path to input images are specified and
# images are loaded with imread command
image1 = cv2.imread('img1.png')          # imrea
image2 = cv2.imread('img2.png')
cv2.imshow("Original Image1",image1)
cv2.imshow("Original Image2",image2)

# cv2.addWeighted is applied over the
# image inputs with applied parameters
weightedSum = cv2.addWeighted(image1, 0.5, image2,
# Syntax: cv2.addWeighted(img1, wt1, img2, wt2, gammaValue)
# Parameters:
# img1: First Input Image array(Single-channel, 8-bit or floating-point)
# wt1: Weight of the first input image elements to be applied to the final image
# img2: Second Input Image array(Single-channel, 8-bit or floating-point)
# wt2: Weight of the second input image elements to be applied to the final image
# gammaValue: Measurement of light

# the window showing output image
# with the weighted sum
cv2.imshow('Weighted Image', weightedSum)
# De-allocate any associated memory usage
```



```python
import cv2
import numpy as np

# path to input images are specified and
# images are loaded with imread command
image1 = cv2.imread('img3.png')
image2 = cv2.imread('img4.png')
cv2.imshow("Original Image:1",image1)
cv2.imshow("Original Image:2",image2)
# cv2.subtract is applied over the
# image inputs with applied parameters
sub = cv2.subtract(image1, image2)

# the window showing output image
# with the subtracted image
cv2.imshow('Subtracted Image', sub)

# De-allocate any associated memory usage
if cv2.waitKey(0) & 0xff == 27:
    cv2.destroyAllWindows()
```

Conclusion:

In this program we have learnt to enhance an image using image arithmetic and logical operations.