Shri Ramdeobaba College of Engineering and Management, Nagpur

Department of Electronics Engineering

Digital Image Processing (ENT 355-3)

Name: Prajwal Pandurang Shette

Roll No: B1- 12

Experiment No: 03

Aim: Python program to perform Mathematical operations on images.

i. Arithmetic Operations

a. Addition , b. Subtraction, c. Multiplication, d. Division

ii. Logical Operation -> NOT, OR, AND

Software Used: Python, Vs Code, Open Cv, Matplotlib

1. Arithmetic Operations

a) Addition

Code:

```
import cv2
import numpy as np

image1 = cv2.imread('image1.jpg')
image2 = cv2.imread('image2.jpg')

weightedSum = cv2.addWeighted(image1, 0.6, image2, 0.5, 0)
cv2.imshow('Weighted Image', weightedSum)

if cv2.waitKey(0) & 0xff == 27:
    cv2.destroyAllWindows()
```

Input Images:

Image1

Image2

Output:

b) Subtraction

Code:

```
import cv2
import numpy as np

image1 = cv2.imread('image1.jpg')
image2 = cv2.imread('image2.jpg')

sub = cv2.subtract(image1, image2)

cv2.imshow('Subtracted Image', sub)

if cv2.waitKey(0) & 0xff == 27:
    cv2.destroyAllWindows()
```

Output:

c) Multiplication

Code:

```
import cv2
import numpy as np

img11 = cv2.imread('image1.jpg')
img22 = cv2.imread('image2.jpg')
result_image = cv2.multiply(img11, img22)
cv2.imshow('Multiply Image', result_image)

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

Output:

d) Division

Code:

```
import cv2
import numpy as np

# Reading image file
img = cv2.imread('image1.jpg')

# Applying OpenCV scalar division on image
fimg = cv2.divide(img, 2)

# Saving the output image
cv2.imwrite('image2', fimg)
```

Output:

ii) **Logical Operation**

a) AND

Code:

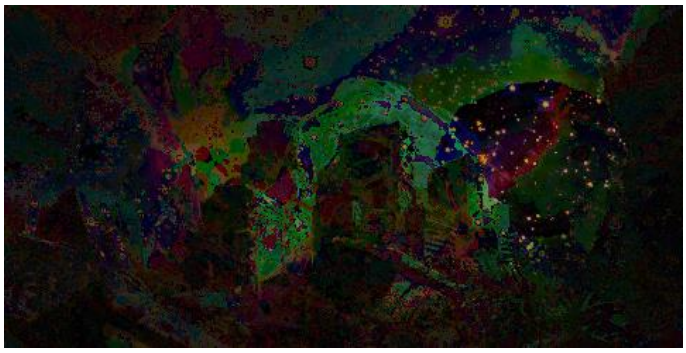
```
import cv2

img1 = cv2.imread("image1.jpg")
img2 = cv2.imread("image2.jpg")
bitwise_and = cv2.bitwise_and(img2, img1)

cv2.imshow("bit_and", bitwise_and)

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

Output:



b) OR

Code:

```
import cv2

img1 = cv2.imread("image1.jpg")
img2 = cv2.imread("image2.jpg")
bitwise_or = cv2.bitwise_or(img2, img1)

cv2.imshow("bitwise_or", bitwise_or)

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

Output:

c) **NOT**

Code:

```
import cv2

img2 = cv2.imread("image1.jpg")
bitwise_not = cv2.bitwise_not(img2)

cv2.imshow("bitwise_not", bitwise_not)

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

Input Image:

Output Image:

Conclusion :

In this Experiment we have learned how to perform different arithmetic operations on images, analyzed the working of different OpenCV methods used for performing image arithmetic, and learned where these image arithmetic operations.