Manthan Gopal Dhole
201080020 IT Ty

Experiment No. 4
Subject: Digital Image Processing

Aim :

Read the images 1.jpg and 2.jpg of equal size. If not resize the images. Perform following image arithmetic operations on it and display the output.

1. Pixel wise Addition (**1.jpg + 2.jpg**)
2. Pixel wise Subtraction (**1.jpg - 2.jpg**)
3. Pixel wise Multiplication (**1.jpg X 2.jpg**)
4. Pixel wise Division (**1.jpg / 2.jpg**)

Theory :

To perform the image arithmetic operations on two images using Python, you can use the OpenCV library. Here are the general steps:

1] Install OpenCV by running the following command in your terminal or command prompt:

```
pip install opencv-python
```

2] Import the library in your Python code:

```
import cv2
```

3] Read the two images using the cv2.imread() function:

4] In python for Addition of images cv2.add(img1,img2),
Subtraction of images cv2.subtract(img1,img2)
Multiplication of images cv2.multiply(img1,img2)
Division of images cv2.division(img1,img2)

```
img1 = cv2.imread('1.jpg')
img2 = cv2.imread('2.jpg')
```

Check if the images are of equal size using the cv2.shape() function. If not, resize them using the cv2.resize() function:

```
if img1.shape != img2.shape:
   img2 = cv2.resize(img2, (img1.shape[1], img1.shape[0]))
```

Perform the pixel-wise arithmetic operations on the two images using the appropriate OpenCV functions:

```
# Addition
add_img = cv2.add(img1, img2)

# Subtraction
sub_img = cv2.subtract(img1, img2)

# Multiplication
mult_img = cv2.multiply(img1, img2)

# Division
div_img = cv2.divide(img1, img2)
```

Display the output images using the cv2.imshow() and cv2.waitKey() functions:

```
cv2.imshow('Addition', add_img)
cv2.imshow('Subtraction', sub_img)
cv2.imshow('Multiplication', mult_img)
cv2.imshow('Division', div_img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Note: Make sure the image files are in the same directory as your Python code or provide the full path to the image files in the cv2.imread() function.

Code :

```
import cv2
```

```
# read the images
img1 = cv2.imread('1.jpg')
img2 = cv2.imread('2.jpg')
img3=  cv2.resize(cv2.imread('3.jpg'),(2448,3264))


# perform pixel-wise addition

def imageAddition():
    addition = cv2.add(img1, img2)

    # resize the output image to a specific size (e.g., 640x480)
    output_size = (640, 480)
    resized = cv2.resize(addition, output_size)

    # display the result
    cv2.imwrite('addition.jpg', resized)

    cv2.imshow('Resized Addition', resized)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
################

def imageSubtraction():
    subtraction=cv2.subtract(img1,img2)
    output_size = (640, 480)

    resized = cv2.resize(subtraction, output_size)

    # display the result
    cv2.imwrite('subtraction.jpg', resized)

    cv2.imshow('Resized Subtraction', resized)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

################
```

```
def imageMultiplication():
    multiplication=cv2.multiply(img1,img2)
    output_size = (640, 480)

    resized = cv2.resize(multiplication, output_size)
    # display the result
    cv2.imwrite('multiplication.jpg', resized)

    cv2.imshow('Resized multiplication', resized)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

#################
def imageDivision():
    division=cv2.divide(img1,img2)
    output_size = (640, 480)


    resized=cv2.resize(division,output_size)
    cv2.imwrite('division.jpg',resized)
    cv2.imshow('Resized Division',resized)
    cv2.waitKey(0)
    cv2.destroyAllWindows()


imageAddition()
imageSubtraction()
imageMultiplication()
imageDivision()
```

Input Images : 1.jpg and 2.jpg

Manthan Gopal Dhole
201080020 IT Ty

Experiment No. 4
Subject: Digital Image Processing

Output Images:

Addition of (1.jpg+2.jpg)

Manthan Gopal Dhole
201080020 IT Ty

Experiment No. 4
Subject: Digital Image Processing

Subtraction of (1.jpg-2.jpg)

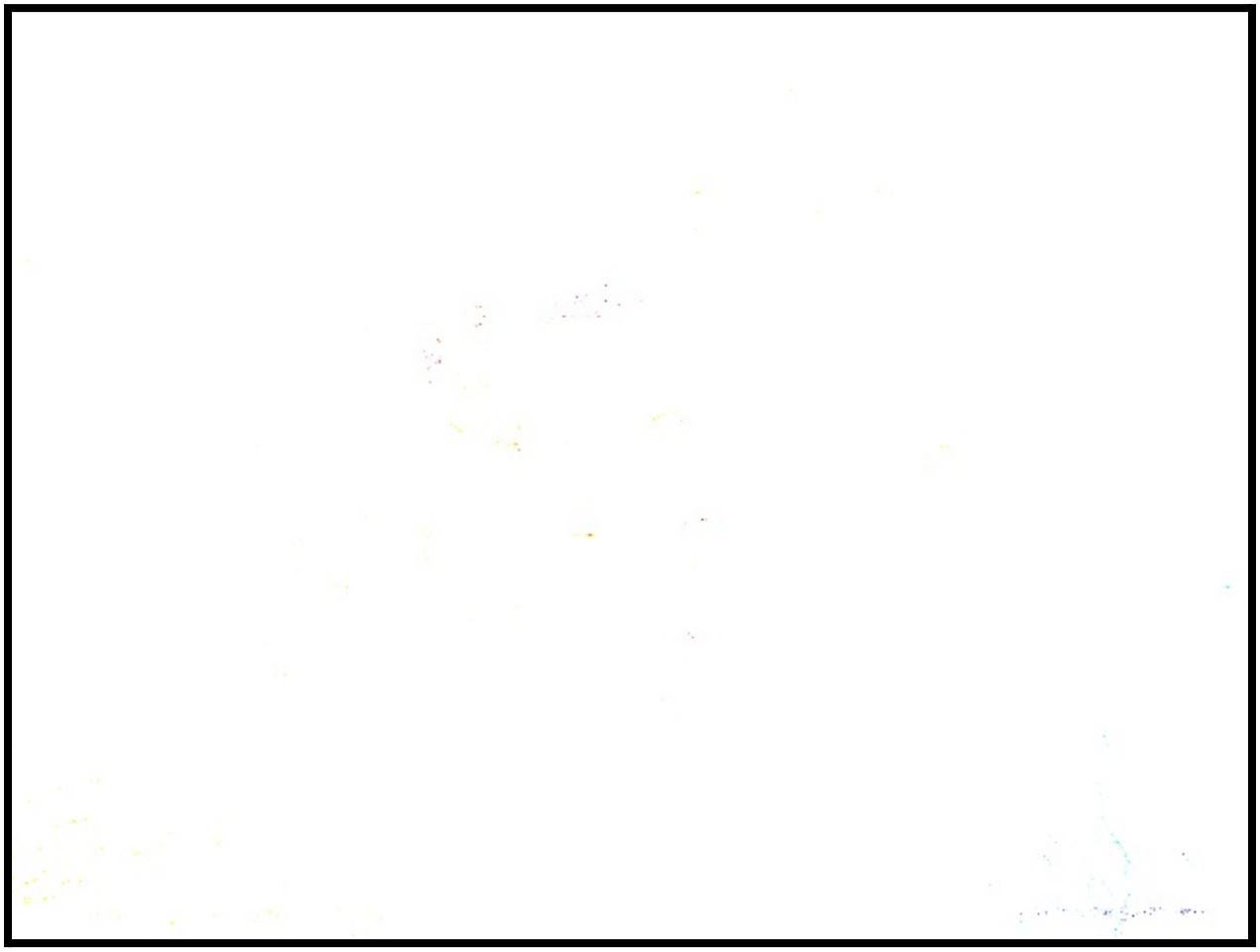
Manthan Gopal Dhole
201080020 IT Ty

Experiment No. 4
Subject: Digital Image Processing

Multiplication of (1.jpg*2.jpg)

Manthan Gopal Dhole
201080020 IT Ty

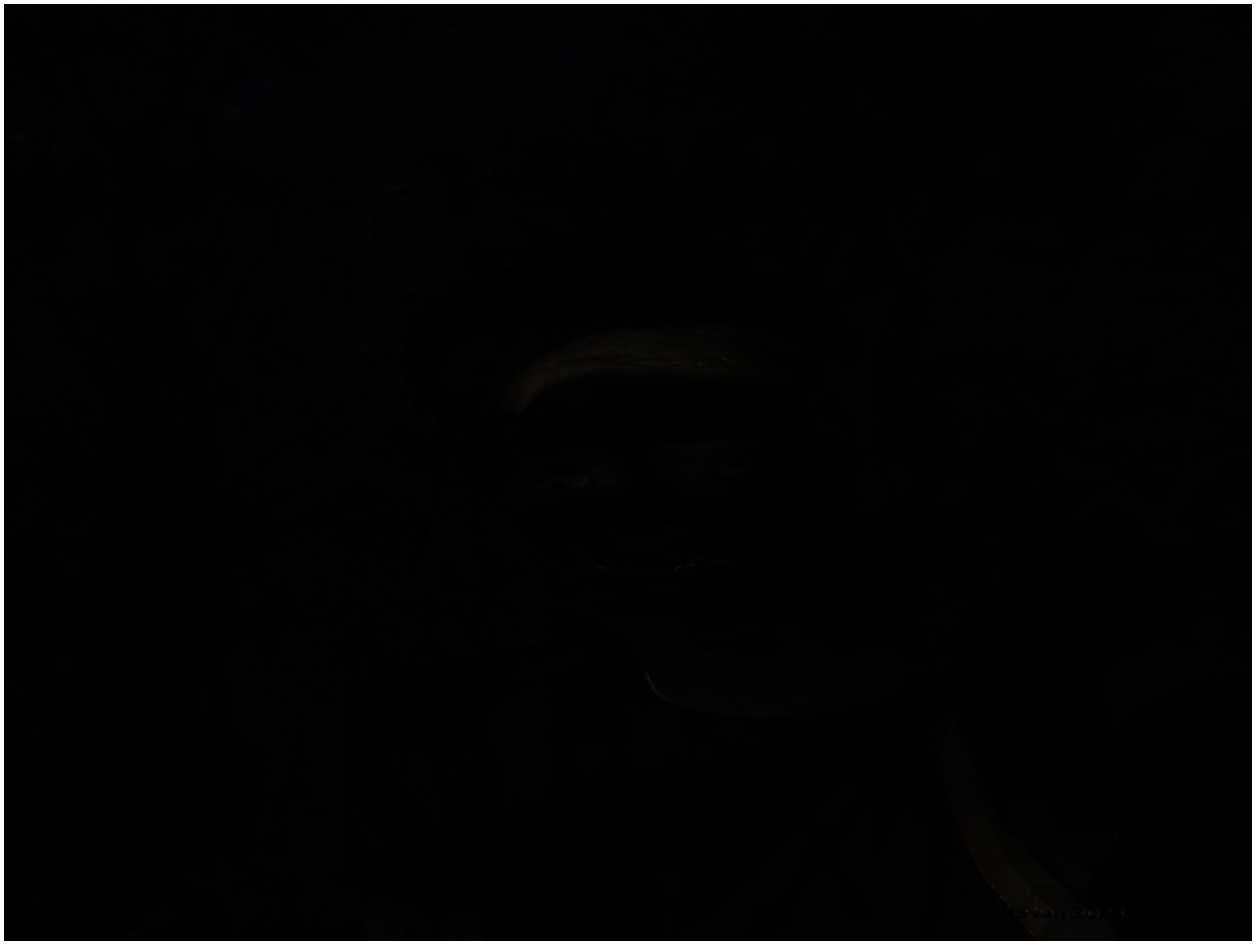
Experiment No. 4
Subject: Digital Image Processing



Division of (1.jpg/2.jpg)

Manthan Gopal Dhole
201080020 IT Ty

Experiment No. 4
Subject: Digital Image Processing



Conclusion :

Image arithmetic operations involve performing mathematical operations on the pixels of two or more images. In this case, we have two images of equal size, and we will be performing four different pixel-wise arithmetic operations on them: addition, subtraction, multiplication, and division.

Addition operation is useful when combining images, such as when creating a composite image or when adding a watermark to an image.

Subtraction operation is useful for detecting differences between two images, such as in medical imaging or in detecting changes in satellite images over time.

Multiplication operation is useful for combining images, such as when creating a composite image or when creating special effects.

Division operation is useful for adjusting the brightness or contrast of an image, as well as for image compression.