**Operations on images:**

As we know from the previous tutorials, Images are stored in the form of matrix in computer. Hence we can perform usual matrix operations on them like addition, subtraction, multiplication and much more. Let’s try some of them:

**1. Addition of two images:**

* For addition of two images, their dimension must be same.
* The datatype of resultant image must be typecast to the original image( usually datatype of images is uint8 ).
* Be careful about the procedure you use to obtain the sum of images.

Let’s compare the results we get on performing different addition procedures.

1. sum1 is obtained by:
2. Divide both the images by 2.
3. Add the two images.
4. Typecast the result to uint8.
5. sum2 is obtained by:
6. Add the two images.
7. Divide the sum we get after adding.
8. Typecast the result to uint8.
9. sum3 is obtained by:
10. Add the two images.
11. Typecast the result to uint8.
12. Divide the result by 2.

Python code:

#import openCV library

import cv2

img1 = cv2.imread('dolphine.jpg',0);

img2 = cv2.imread('water.jpg',0);

print(img1.shape);

print(img2.shape);

# img1.shape: (183, 275)

# img2.shape: (612, 612)

# now crop images to make their dimensions same

img1 = img1[0:183,0:275];

img2 = img2[0:183,0:275];

cv2.imshow('image1',img1);

cv2.imshow('image2',img2);

# one variant of addition

sum1 = img1/2 + img2/2;

sum1 = sum1.astype('uint8');

#second variant of addition

sum2 = (img1+img2)/2;

sum2 = sum2.astype('uint8');

#third variant of addition

sum3 = img1+img2;

sum3 = sum3.astype('uint8')/2;

cv2.imshow('sum1',sum1);

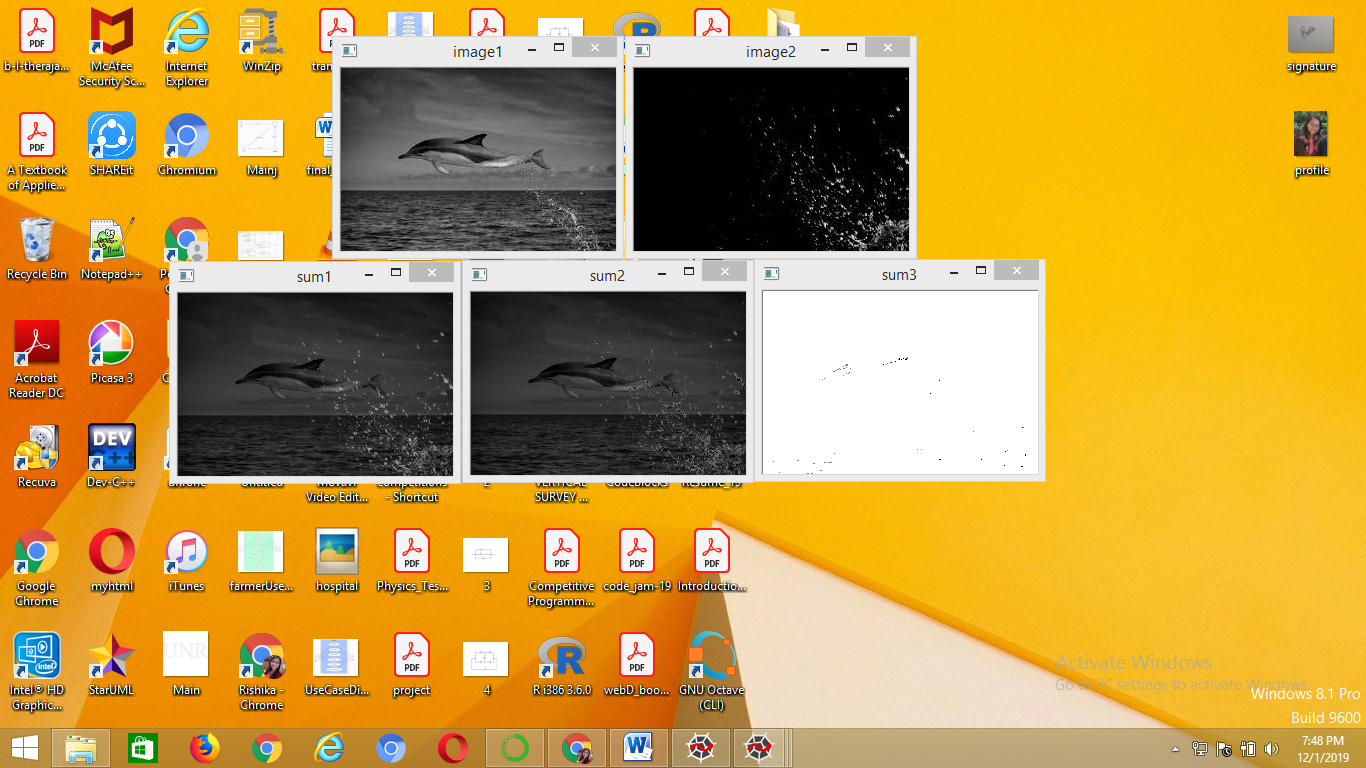
cv2.imshow('sum2',sum2);

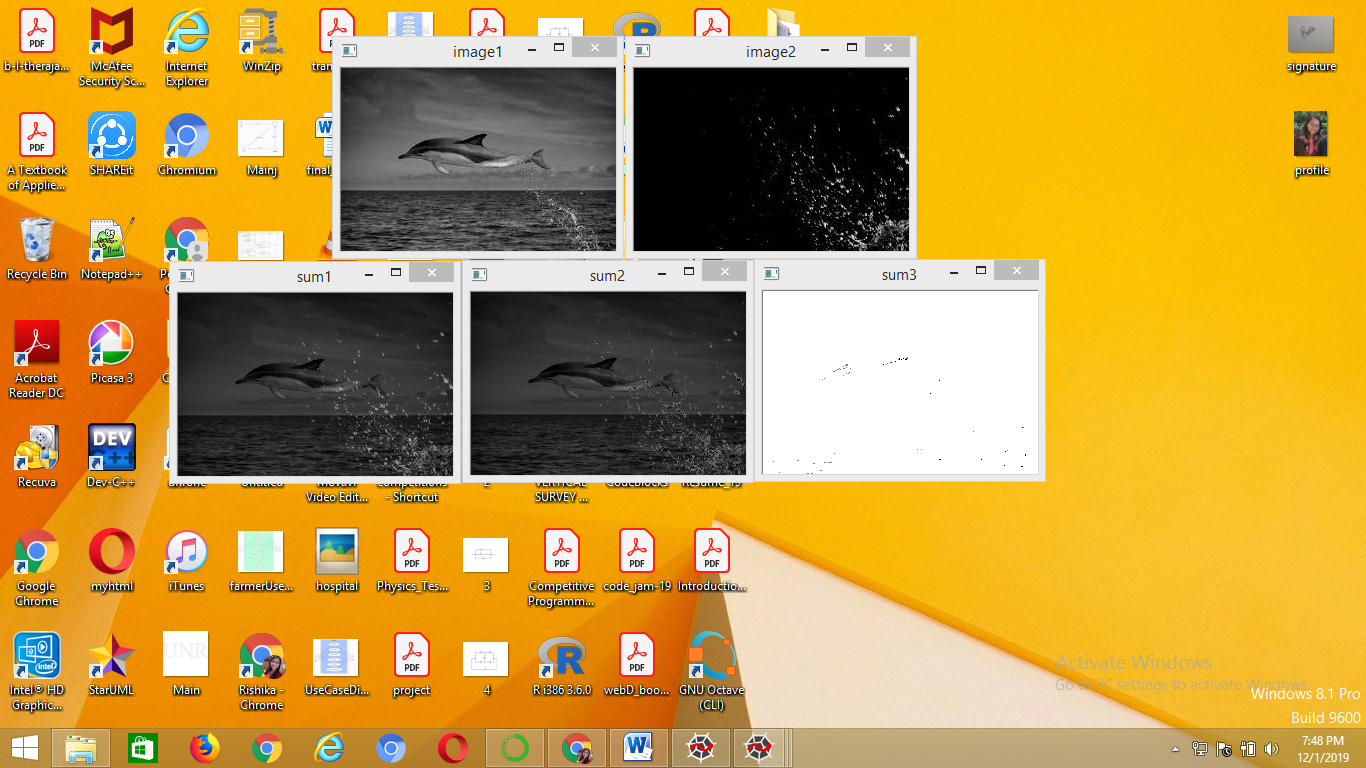
cv2.imshow('sum3',sum3);

cv2.waitKey(30000);

cv2.destroyAllWindows();

Output:





The difference in all the images come because of typecasting as all the numbers greater than 255 is converted to 255 due to integer overflow.

**Multiply an image by a scalar:**

Multiplying an image by a scalar will increase the brightness of images. Let’s try it:

Python code:

#import openCV library

import cv2

img1 = cv2.imread('flower.jpg',0);

img2 = 2.\*img1;

img2 = img2.astype('uint8');

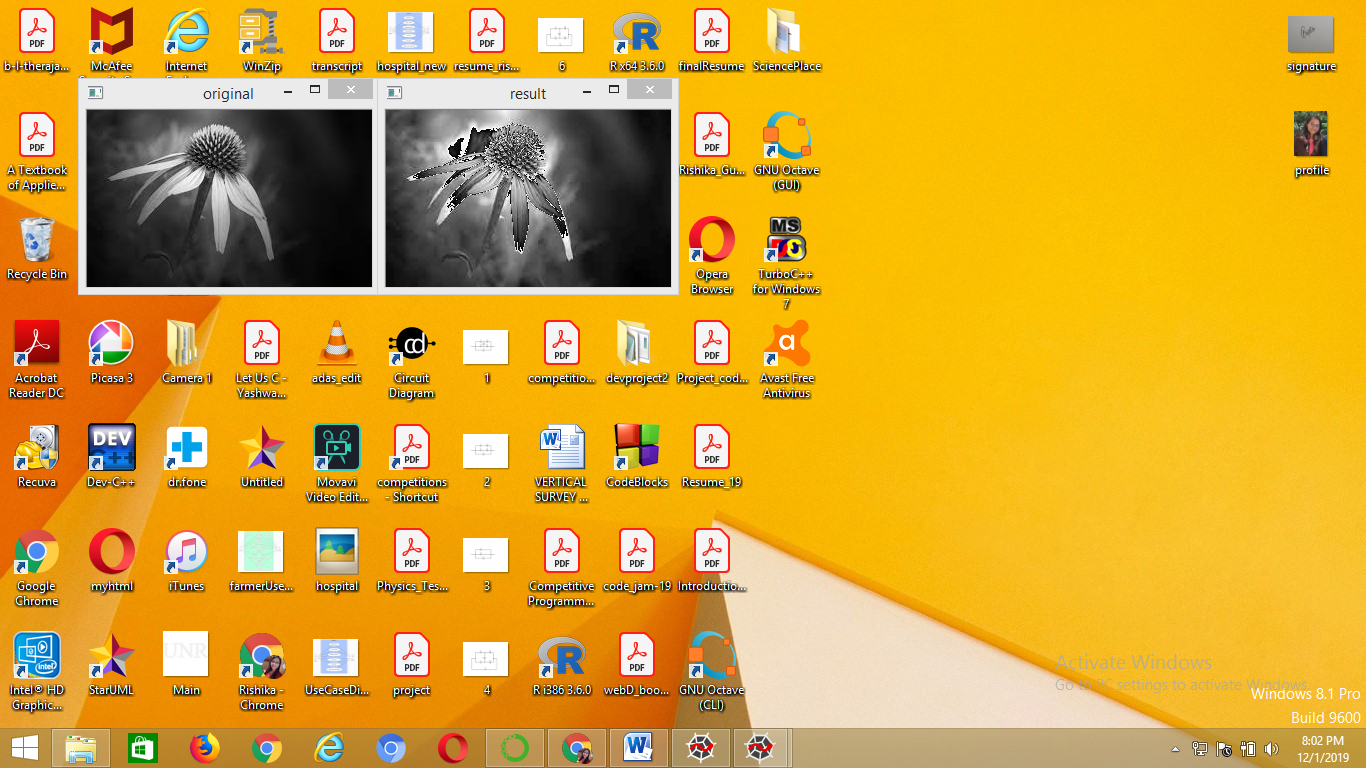
cv2.imshow('original',img1);

cv2.imshow('result',img2);

cv2.waitKey(30000);

cv2.destroyAllWindows();

Output:



**Subtraction between images:**

* For subtraction of two images, their dimension must be same.
* The datatype of resultant image must be typecast to the original image( usually datatype of images is uint8 ).
* Be careful about the procedure you use to obtain the difference between two images.

Let’s compare the results we get on performing different subtraction procedures.

1. diff1 is obtained by:
2. Subtract img2 from img1.
3. Typecast the result to uint8.
4. sum2 is obtained by:
5. Subtract img1 from img2.
6. Typecast the result to uint8.
7. sum3 is obtained by:
8. Add the absolute difference between img1 and img2, and img2 and img1.
9. Typecast the result to uint8.

Python code:

#import openCV library

import cv2

img1 = cv2.imread('dolphine.jpg',0);

img2 = cv2.imread('water.jpg',0);

print(img1.shape);

print(img2.shape);

# img1.shape: (183, 275)

# img2.shape: (225, 225)

# now crop images to make their dimensions same

img1 = img1[0:183,0:225];

img2 = img2[0:183,0:225];

cv2.imshow('image1',img1);

cv2.imshow('image2',img2);

# one variant of subtraction

diff1 = img1 - img2;

diff1 = diff1.astype('uint8');

#second variant of subtraction

diff2 = img2 - img1;

diff2 = diff2.astype('uint8');

#third variant of subtraction

diff3 = (img1-img2).astype('uint8')+(img2-img1).astype('uint8');

diff3 = diff3.astype('uint8');

cv2.imshow('diff1',diff1);

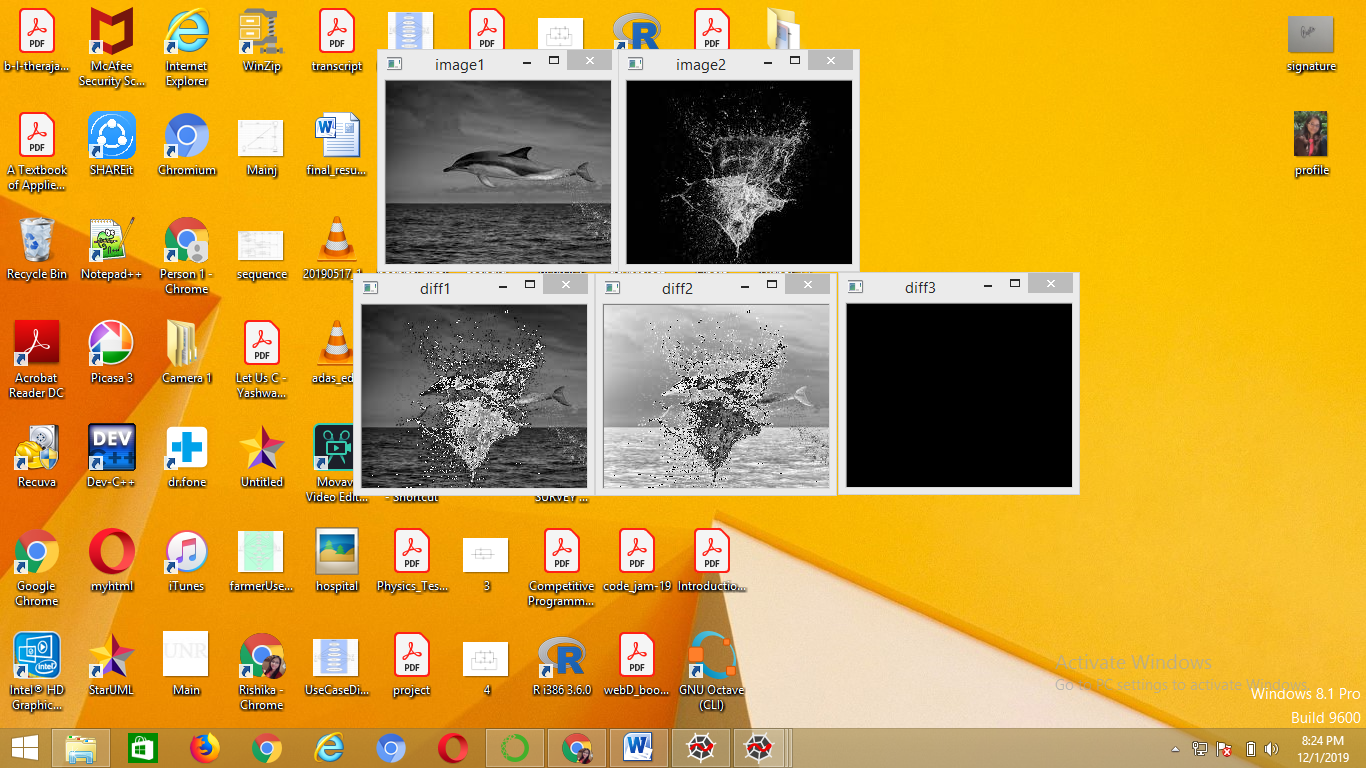
cv2.imshow('diff2',diff2);

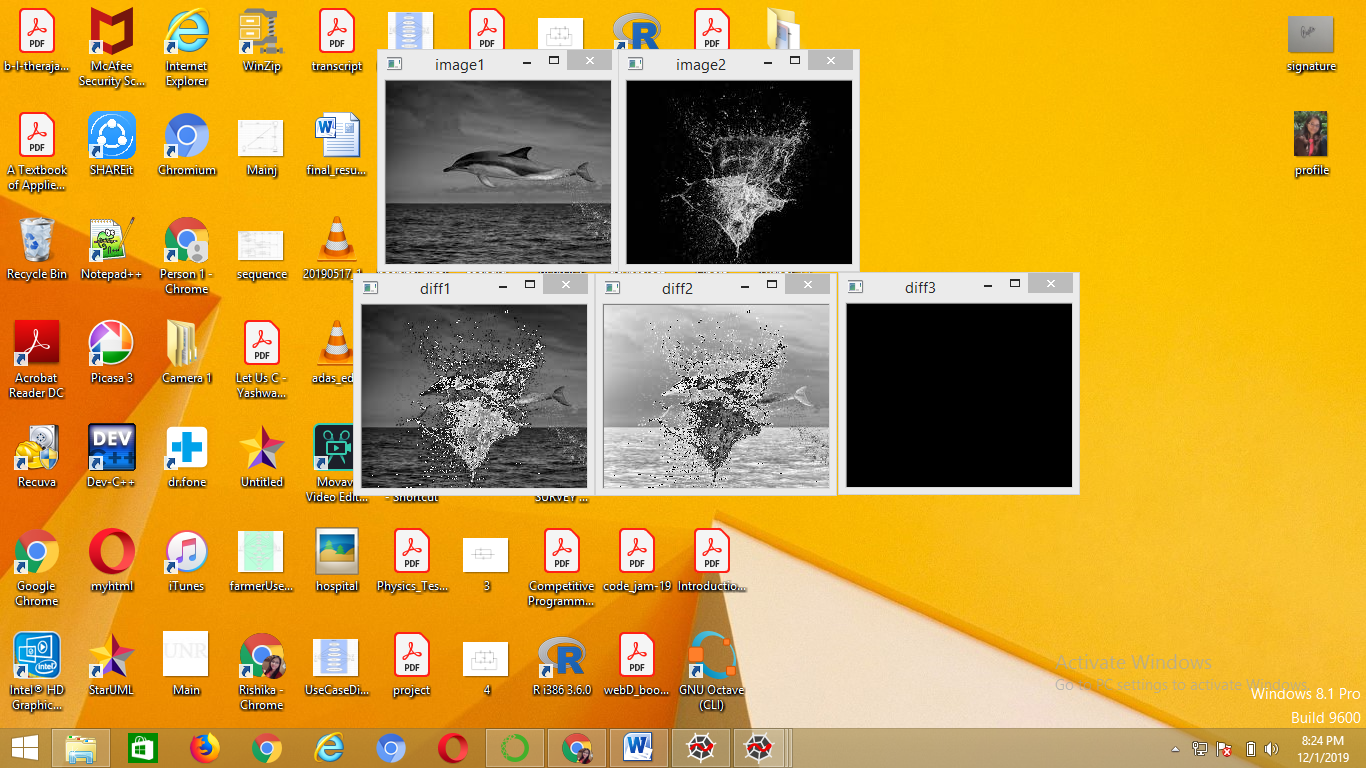
cv2.imshow('diff3',diff3);

cv2.waitKey(30000);

cv2.destroyAllWindows();

Output:





The difference in all the images come because of typecasting as all the numbers less than 0 is converted to 0 due to the datatype of image as unsigned int.