**Image processing using openCV: Scaling**

Scaling simply refers to resizing of images.

cv2.resize(image, dsize, fx, fy, interpolation) function can be used to resize the image down to or up to the desired size. It has the following arguments:

1. image: It specifies the image whose size has to be changed.
2. dsize: It is the desired size of the image.
3. fx: Horizontal scaling factor.
4. fy: Vertical scaling factor.
5. interpolation: flag used for different scaling technique. It has five variations:
6. INTER\_NEAREST: a nearest-neighbor interpolation.
7. INTER\_CUBIC: it is slow but enlarge image look best with this.
8. INTER\_LINEAR: faster than INTER\_CUBIC but result is not as good as in INTER\_CUBIC.
9. INTER\_AREA: best for image shrinking.
10. INTER\_LANCZOS4: a lanczos interpolation over 8×8 pixel neighborhood.

Resizing of image can be done both by preserving and not preserving the aspect ratio of the image.

**Resizing with preserving the aspect ratio**

Python code:

import cv2

img = cv2.imread('flower.jpg',1)

scaling\_factor = 60 # as it is <100, hence it will scale down the image

width = int(img.shape[1] \* scaling\_factor / 100);

height = int(img.shape[0] \* scaling\_factor / 100);

dim = (width, height);

# resize the image

resized = cv2.resize(img, dim, interpolation = cv2.INTER\_AREA);

print('Original Dimensions : ',img.shape);

print('Resized Dimensions : ',resized.shape);

#Original Dimensions : (177, 285, 3)

#Resized Dimensions : (106, 171, 3)

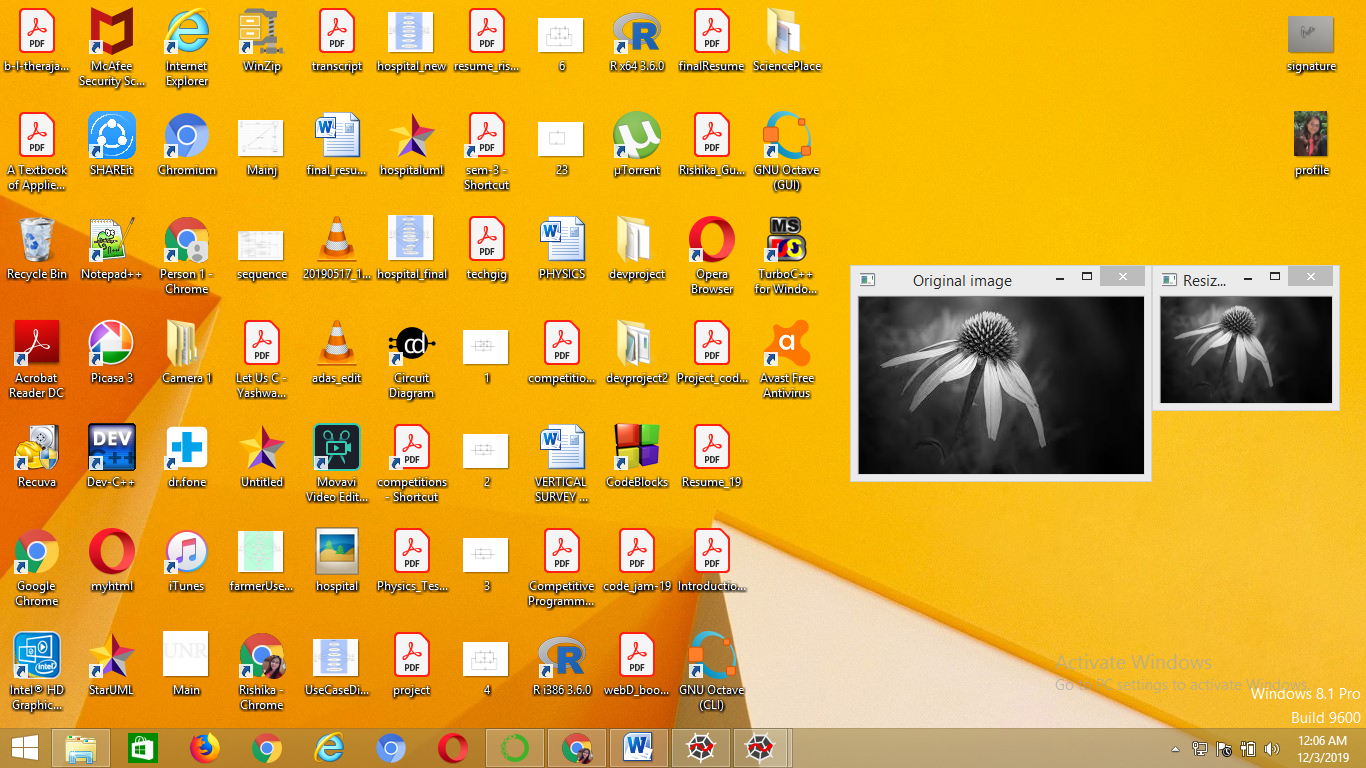
cv2.imshow("Original image", img);

cv2.imshow("Resized image", resized);

cv2.waitKey(0);

cv2.destroyAllWindows();

Output:



**Resizing without preserving the aspect ratio**

Python code:

import cv2

img = cv2.imread('flower.jpg',1)

fx = 60 # horizontal scaling factor

fy = 120 # vertical scaling factor

width = int(img.shape[1] \* fx / 100);

height = int(img.shape[0] \* fy / 100);

dim = (width, height);

# resize the image

resized = cv2.resize(img, dim, interpolation = cv2.INTER\_AREA);

print('Original Dimensions : ',img.shape);

print('Resized Dimensions : ',resized.shape);

#Original Dimensions : (177, 285, 3)

#Resized Dimensions : (212, 171, 3)

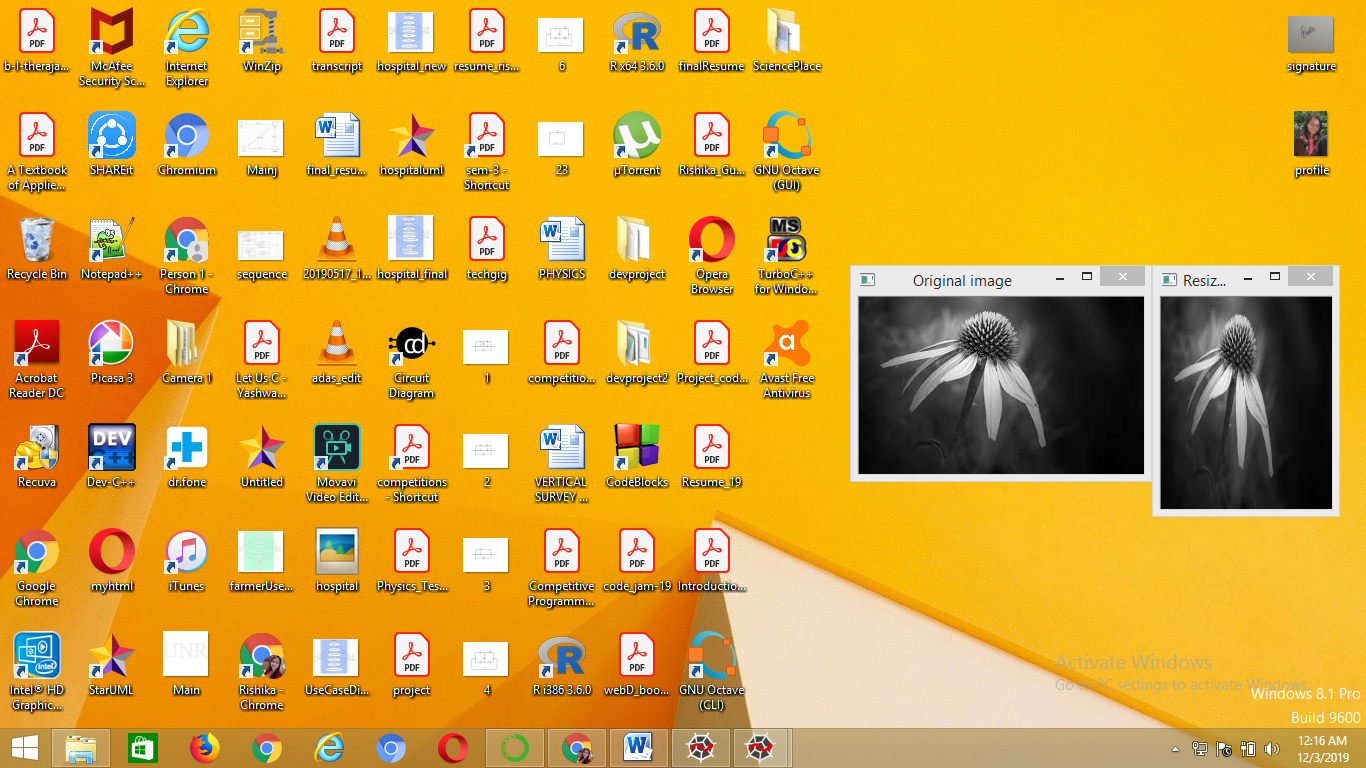
cv2.imshow("Original image", img);

cv2.imshow("Resized image", resized);

cv2.waitKey(0);

cv2.destroyAllWindows();

Output:



We can also specify the width and height of the image by explicitly passing the desired size.

For example:

Python code:

import cv2

img = cv2.imread('flower.jpg',1)

fx = 477 # desired width

fy = 200 # desired height

dim = (fx, fy);

# resize the image

resized = cv2.resize(img, dim, interpolation = cv2.INTER\_AREA);

print('Original Dimensions : ',img.shape);

print('Resized Dimensions : ',resized.shape);

#Original Dimensions : (177, 285, 3)

#Resized Dimensions : (212, 171, 3)

cv2.imshow("Original image", img);

cv2.imshow("Resized image", resized);

cv2.waitKey(0);

cv2.destroyAllWindows();

Output:

