**Image smoothing techniques:**

As we have seen, the images obtained after performing operations like masking, segmentation etc. also contain noise which is generally not desirable in image processing as it creates disturbance in the output. Hence it becomes necessary to remove noise from the image.

Image smoothing removes high frequency content like noise, edges etc. and make them blur.

There are mainly four techniques to smooth the image in OpenCV:

1. Averaging:

It is done by changing every pixel information to the average of its neighbouring pixels in the defined area( known as kernel area ).

cv2.blur( source, kernel shape ) function is used to perform averaging in openCV. It has two arguments:

1. source: it signifies the source image on which smoothing has to be done.
2. kernel shape: it is a tuple of type (width, height) to signifies the shape of kernel area.

It returns the image after performing averaging on the source image.

Python code:

# Importing cv2

import cv2

#read input image

img = cv2.imread('dance.jpg');

#increasing the kernel size will increase the blurring effect

blur1 = cv2.blur(img,(5,5));

blur2 = cv2.blur(img,(10,10));

blur3 = cv2.blur(img,(20,20));

cv2.imshow('Original',img);

cv2.imshow('blurred image 5x5',blur1);

cv2.imshow('blurred image 10x10',blur2);

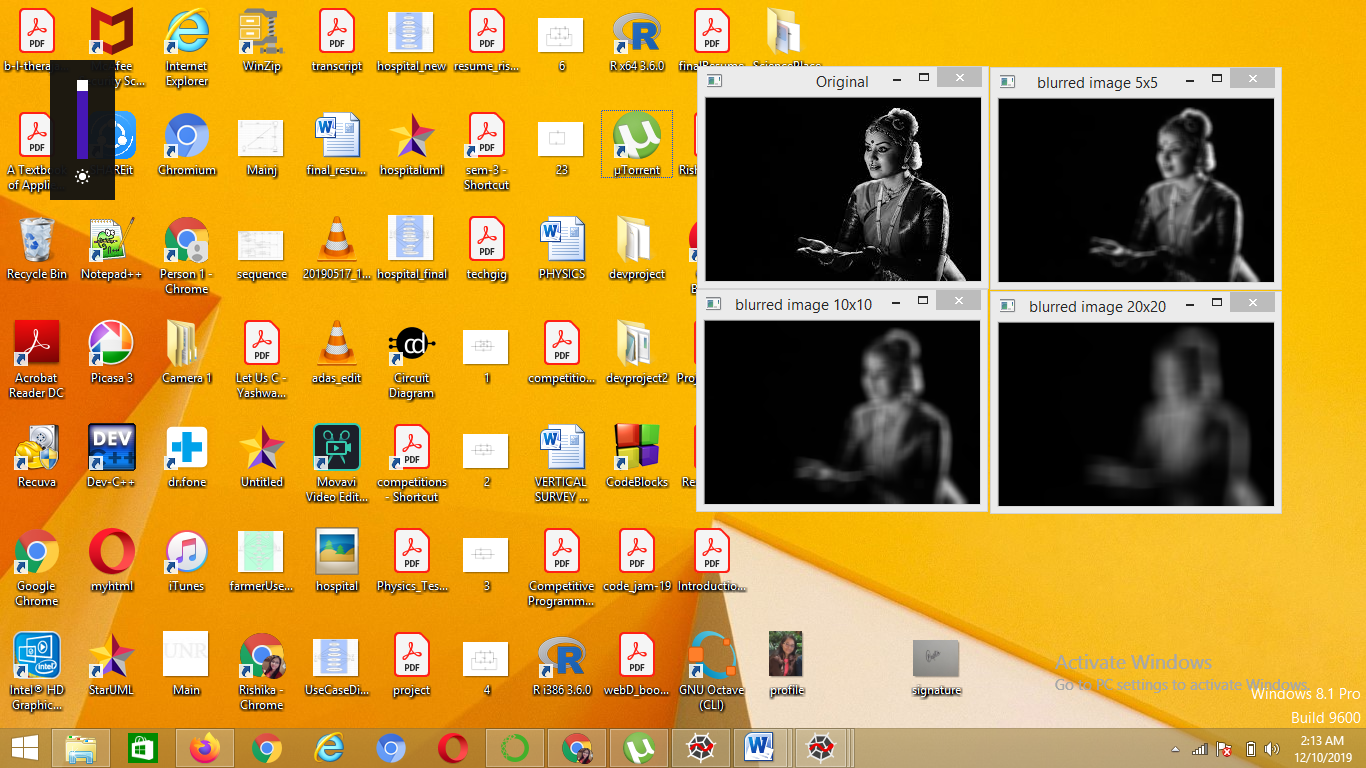
cv2.imshow('blurred image 20x20',blur3);

#wait for 10 sec

cv2.waitKey(10000);

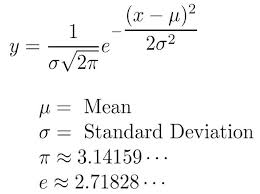
cv2.destroyAllWindows();

Output:

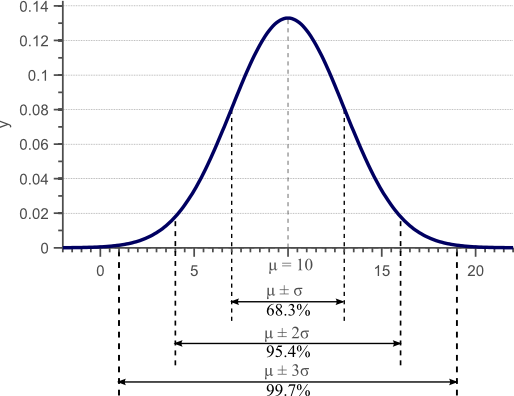


1. Gaussian Blurring:

In this method, Gaussian function is used to produce the kernel matrix for noise reduction. Gaussian function is defined as



Plot of Gaussian function:



cv2.GaussianBlur( source, kernel shape, x\_dev, y\_dev) function is used in openCV to perform gaussian blur. The function takes four arguments:

1. source: it signifies the source image on which smoothing has to be done.
2. kernel shape: it is a tuple of type (width, height) to signifies the shape of kernel area where width and height should be odd.
3. x\_dev: it signifies the standard deviation in x direction.
4. y\_dev: it signifies the standard deviation in y direction.

It returns the image after performing Gaussian blur on the source image.

Notes:

* If y\_dev is not passed in the function then y\_dev will be taken same as x\_dev.
* If both are given as zeros then they are calculated from the kernel size.

Python code:

# Importing cv2

import cv2

#read input image

img = cv2.imread('dance.jpg');

blur1 = cv2.GaussianBlur(img,(5,5),1,2);

blur2 = cv2.GaussianBlur(img,(5,5),4);

blur3 = cv2.GaussianBlur(img,(5,5),0);

cv2.imshow('Original',img);

cv2.imshow('blurred image 1',blur1);

cv2.imshow('blurred image 2',blur2);

cv2.imshow('blurred image 3',blur3);

#wait for 10 sec

cv2.waitKey(10000);

cv2.destroyAllWindows();

Output:

