**CS 512 – ASSIGNMENT 2**

**CWID – A20377210**

1)

a)

**Signal to Noise Ratio in an Image:**

The signal-to-noise ratio (SNR) is used in imaging as a physical measure of the [sensitivity](https://en.wikipedia.org/wiki/Sensitometry) of image.

SNR is measured using decibels of power by applying 10 log rule to the actual SNR ratio.

The formula to calculate SNR can be given as,

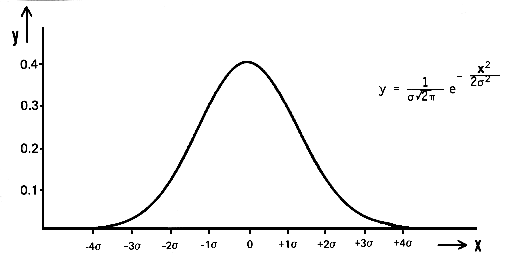
SNR=10log10(Psignal/Pnoise) where, Psignal=Mean of pixel value and Pnoise=Standard

deviation or the error value of the pixel

b)

**Gaussian vs. Impulsive Noise:**

Gaussian noise is a random variable with Gaussian probability distribution function(pdf) which can be written as,

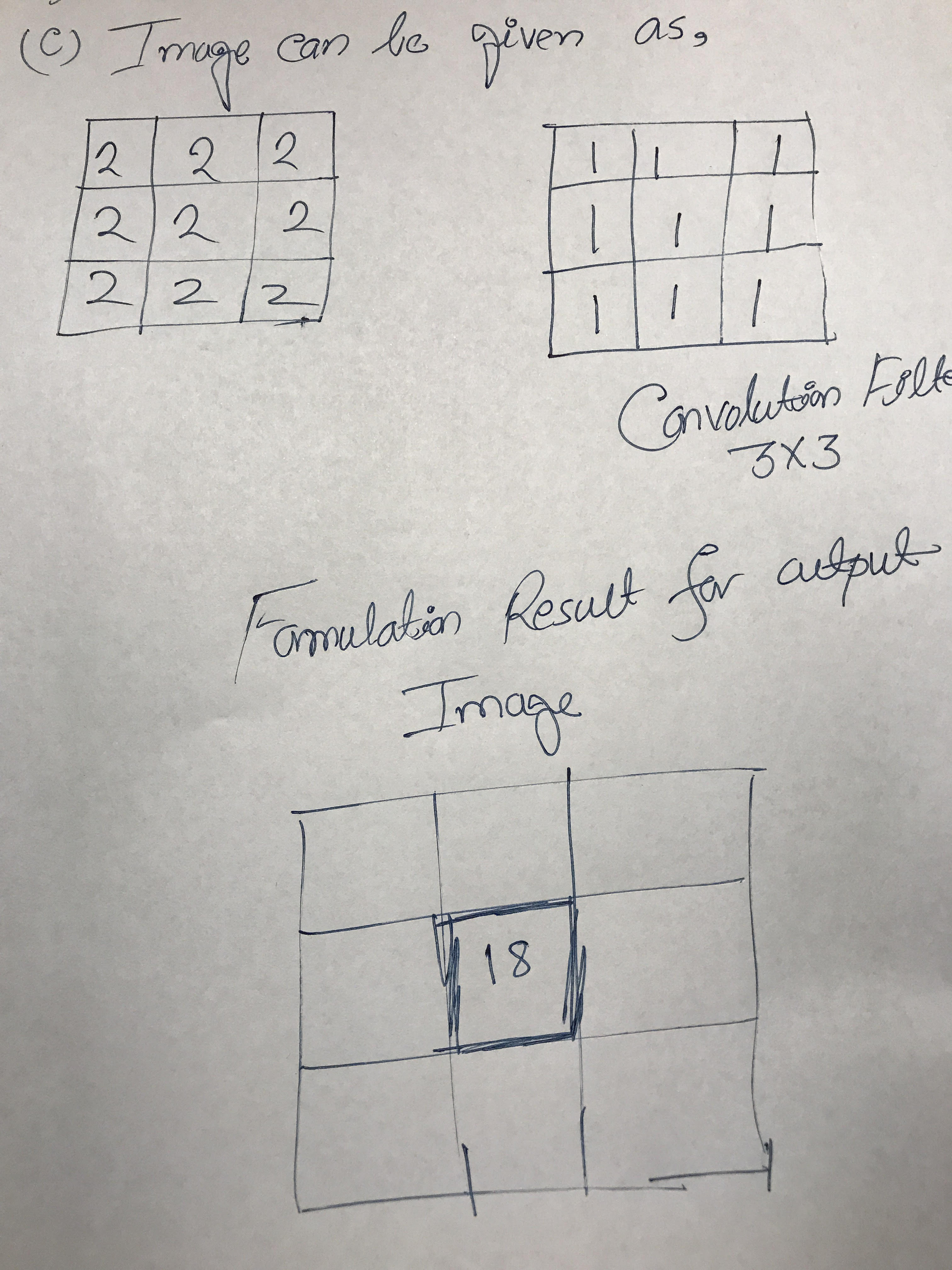


Whereas impulse noise is a peak that can be a random value as well as any value arises from transmission error.

**Median Filter vs. Mean Filter:**

The median filter is much less sensitive to impulsive noise than the mean filter. So median filters do better job when it comes to remove noise compare to mean filters.

c)



d)

In order to find the derivative of an image convolved with a filter we can make it more efficient by the following derivative property.

d/dx(f\*g)=(df/dx)\*g=f\*(dg/dx)

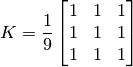
e)

**Handle Boundaries in Convolution:**

1. By applying a padding of a constant value for the whole border. This value will be updated randomly each 0.5 seconds.
2. The border will be replicated from the pixel values at the edges of the original image.
3. Only look at the result values and ignore boundaries.

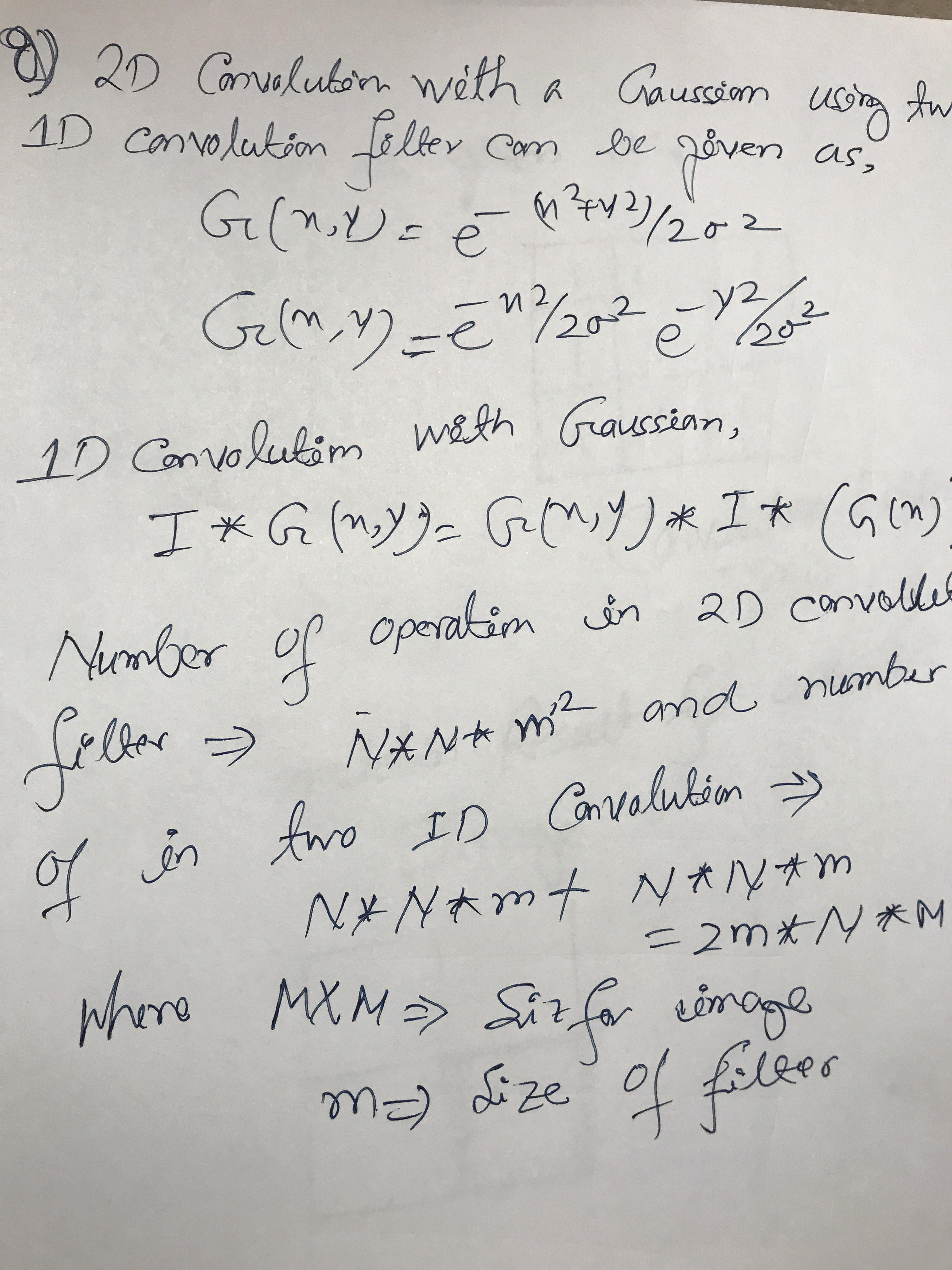
f)

A basic 3x3 smoothing filter can be given as below which is also known as a box filter.



Here we can see that the sum of all elements are equal to 9 as because It simply takes the average of all the pixels under kernel area and replaces the central element with this average.

g)



h)

mu=2

size of the filter, m>=5\*mu

m>=5\*2=10

i)

**Gaussian Image Pyramid:**

To form a Gaussian pyramid, subsequent images are weighted down using a Gaussian average ([Gaussian blur](https://en.wikipedia.org/wiki/Gaussian_blur)) and scaled down. Each pixel containing a local average that corresponds to a pixel neighborhood on a lower level of the pyramid.

**Usefulness of Image Pyramids:**

Normally, we used to work with an image of constant size. But in some cases , we need to work with images of different resolution of the same image. For example, while searching for something in an image, like face, we are not sure at what size the object will be present in the image. In that case, we will need to create a set of images with different resolution and search for object in all the images. These set of images with different resolution are called Image Pyramids.

j)

**Laplacian Pyramid:**

Laplacian pyramid is nothing but the difference between up-sampled Gaussian pyramid level and Gaussian pyramid level.

Below are steps used to create a Laplacian Pyramid,

Image->Smoothing->Sub-sampling by 2->up sampling by 2->Smoothing

Laplacian Pyramids are used to reconstruct an up-sampled image from an image lower in the pyramid (with less resolution).

2)

a)

**Edge Detection:**

Edge detection is done based on the change of image gradient value for an image. The purpose of using edge detection is to get the sharp changes in image brightness which will help us to capture all important events in an image. Applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image.

**Properties:**

Below are the main properties one should observe while working with edge detection.

1. Discontinuities in depth
2. Discontinuities in surface orientation
3. Changes in material properties
4. Variations in scene illumination

b)

**Edge Detection Steps:**

**i)Smoothing:**

Sinceedge detection is susceptible to noise in the image so first step is to remove the noise in the image with a Gaussian filter or using any other smoothing technique.

**ii) Enhancement:** Apply a filter to enhance the quality of the edges in the image (sharpening).

**iii) Detection:**

Smoothened image is then filtered with a Sobel kernel in both horizontal and vertical direction to get

 first derivative in horizontal direction and vertical direction. From these two images, we can then find edge gradient and direction for each pixel. After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge.

**iv) Localization:** determine the exact location of an edge (sub-pixel resolution might be required for some applications, that is, estimate the location of an edge which is better than the spacing between pixels). Edge thinning and linking are usually required in this step.

c)

**Image Gradient Filters:**

Below given two filters which are used to find image gradient.

**i) The Sobel Derivatives:**

It computes an approximation of the gradient of an image intensity function. The Sobel Operator combines Gaussian smoothing and differentiation.

**ii)The Laplacian Derivatives:**

It calculates the Laplacian of the image and then the created kernel is used for filtering.

**Image Gradient:**

An **i**mage gradient is a directional change in the intensity or color in an image. Each pixel of a gradient image measures the change in intensity of that same point in the original image, in a given direction. The length of the gradient vector corresponds to the rate of change in that direction.

**Usefulness:**

Image gradients can be used to extract information from images. The magnitude of gradient provides information about the strength of the edge. After gradient of images have been computed, pixels with large gradient values become possible edge pixels.

d)

**Sobel Filter from Smoothing and Derivative Filter:**

The Sobel operator uses two 3x3 kernels  which … are [convolved](https://en.wikipedia.org/wiki/Kernel_(image_processing)#Convolution) with the original image to calculate approximations of the [derivatives](https://en.wikipedia.org/wiki/Image_Derivatives) – one for horizontal changes (Gx), and one for vertical (Gy).After getting Gx and Gy, at each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude and direction, using

G=sqrt(Gx^2+Gy^2) and the direction can be calculated as, Theta=atan(Gy/Gx)

The Sobel Operator also combines Gaussian smoothing and differentiation.

f)

**Edge Detection Using Derivatives:**

An image is a 2D function, so operators describing edges are expressed using partial derivatives.

Points which lie on an edge can be detected by:

(1) detecting local maxima or minima of the first derivative

(2) detecting the zero-crossing of the second derivative

**g) The Laplacian-of-Gaussian (LOG):**

If mu=1, then the LOG can be given as,

G (x, y) = e ^−( x 2+y 2)/2mu^2= e ^−( x 2+y 2)/2

**Detect Edges Using LOG:**

First to reduce the noise effect, the image is smoothed with a low-pass filter.

In the case of the LOG, the low-pass filter is chosen to be a Gaussian which can be given as,

G (x, y) = e ^−( x 2+y 2)/2mu^2

Where mu determines the degree of smoothing, mask size increases with mu.

h)

**Canny Edge vs. Other Edge Detection Algorithms:**

Canny Edge Detection is considered to be a better edge detection than others due to below steps:

1. The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges
2. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum
3. A third criterion is to have only one response to a single edge.

Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. According to Canny edge detection, edge candidates which are not dominant in their neighborhood aren't considered to be edges.

i)

The Non Maximum Suppression and Hysteresis Process for Canny edge detection can be given as,

1. Non Maximum Suppression - Edges candidates which are not dominant in their neighborhood aren't considered to be edges.
2. Hysteresis Process - While moving along the candidates, given a candidate which is in the neighborhood of an edge the threshold is lower.

These two steps reduce the number of "False" edges in an image.