Assignment 2 Answers

1 a) Signal to noise ratio (SNR) in an image is estimated as the ratio between variance of pixels in the sequence of images and variance in the uniform area of the image i.e,

$$SNR = \frac{\text{Variance of pixels in the sequence of images}}{\text{Variance in the uniform area of the image}}$$

Which is $\frac{\text{Variance of Signal}}{\text{Variance of noise}} = \frac{E_s}{E_N} = \frac{\sigma_s^2}{\sigma_N^2}$

b) Gaussian noise follows normal distribution i.e $N(\mu, \sigma^2)$ where μ is mean & σ^2 is the variance. the intensity of the noise decreases as the distance from mean increases

Impulsive noise doesn't have a pattern. The noise can achieve its high or low in any point of time.

Median filter better handles the impulsive noise as it is not affected by peaks, whereas average filter is affected by peaks.

c) $$\begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} = 2 \times 9 \times \frac{1}{9} = 2$$

Value of pixels in the image after applying Convolution is 2

d) It would be more efficient to take the derivative of the filter and convolve it with the image as this would be computationaly efficient and yields the same result.

e) 3 different ways of handling boundaries during Convolution

1- Zero padding : assumes that the pixel value of the image is 0 for those corresponding places when the filter goes out of the image boundary.

2. Replication : assumes that the pixel value of the image is same as its neighbor.

3- ignore boundaries : doesn't care about the convolution results for the boundaries.

f)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

; Sum of all entries is 1. Sum is chosen to be 1 to hold the intensity of pixels in the original image. if not for 1 the intensity of the pixels may change.

g) Gaussian is seperable :

$$\text{i.e., } G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}} \cdot e^{-\frac{y^2}{2\sigma^2}}$$

$$= G(x) \cdot G(y)$$

$$\Rightarrow I * G(x,y) = G(y) * (I * G(x))$$

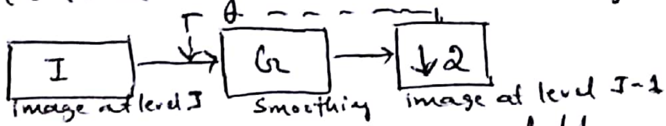
which means, the Image can be convolved with any of the 1D filter and the result is convolved with the other 1D filter. And this gives the same result. and is more efficients as this involves $2m \times M \times N$ operations as opposed to $M \times N \times m^2$ operation when using 2D filter, wher $M \times N$ is size of image & m is size of the filters.

h) given 1D filter, with $\sigma = 2$

m, size of filter, should be $\geq 2\sigma$

$\Rightarrow$ m should be 11

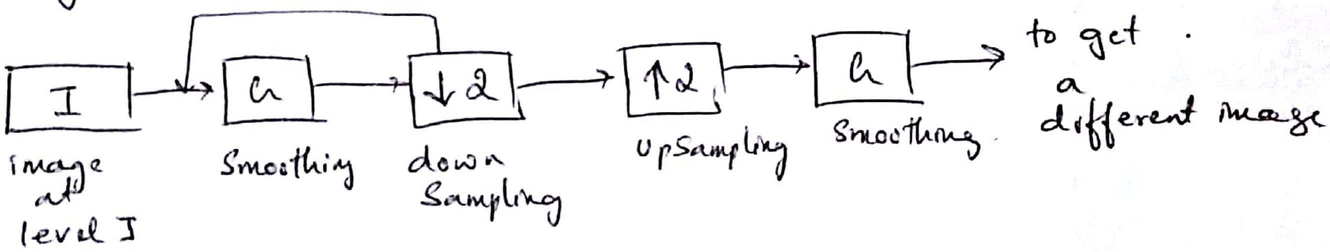
i) Gaussian image pyramid is produced by the MultiScale Analysis of the image. image at a next/different level are produced by sampling from the image at this level. We move from higher resolution to lower resolution by Under sampling by a factor of 2.



Reason for this is to analyse the image in different Scale/resolution.

The additional processing introduced by this is surprizingly only about $\frac{1}{3}$ of the computation of processing involved in the processing of single image.

j) Additionall to the guassian image pyramid, the Laplacian pyramid is produced by upsampling the ~~doo~~ under sampled image, by duplicating the the neighbors. And the resulting image is smoothed again to get a different image.



the Use of this method is that the resulting image shows the error introduced by undersampling step.

2 a) Edge detection is useful because it tells the Change in the image or identifies the discontinuity of the normal

desired properties of edge detection :

1. correspond to the scene elements
2. Should be invariant to point of view, illumination
3. Should ~~and~~ be reliable and consistent.

b) Basic steps of edge detection :

1. Smoothing, without affecting edges → this step is helpful in finding out the region where the edges are present and ignoring other regions.
2. Enhance edges → this step is helpful in pronouncing the edges.
3. Localize edge → this step is helpful in knowing where exactly the edge is.

c) two filters that can be used for computing the image gradient are :

1. Forward difference, by considering the difference between next pixel and the current pixel.

$$I_x = I(x+1, y) - I(x, y) \quad \& \quad \Delta I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

$$I_y = I(x, y+1) - I(x, y)$$

2) Central difference, by considering the difference between next pixel and the previous pixel

$$I_x = I(x+1, y) - I(x-1, y)$$

$$I_y = I(x, y+1) - I(x, y-1)$$

Image gradient is the derivative of the image in the x & X axis.

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$$

I_x : how much image changes in x direction

I_y : how much image changes in y direction.

Image gradient, gives vectors that are perpendicular to edges in the image. This can be used in the detection of edge, if the gradients are in different direction, there is an edge.

d) Sobel filter can be produced by convolving the Smoothing & derivative filters along that axis.

Sobel x is produced by $G * \frac{\partial}{\partial x}$

Sobel Y is produced by $G * \frac{\partial}{\partial Y}$

e) More accurate derivative filter can be calculated by the 1) perfect reconstruction of the orginal image from the sampled image.
2) taking derivative of the orginal image
3) then sampling from the derivative result of the image

Additionally, we can use sinc function (h(x)) or approximation of it to Gaussian (G(x)).

$$h(x) = \frac{\sin(\pi x/T)}{(\pi x/T)}$$

f) We can find the region where the edge is pronounced by computing the first derivative, i.e., the change in the direction of gradient vector.

By taking the 2nd derivative over the pronounced area we can localize the edge at the zero crossing.

g) $\sigma = 1$, $LoG = (r^2 - 2)\, e^{-\frac{r^2}{2}}$

1. Compute Laplacian of Gaussian, ΔG
2. Convolve with image $I * \Delta G$.
3. have a threshold z

   mark pixel as 1 if $I * \Delta G > z$

   0 otherwise

4. detect edges at transition from $0 \to 1$ or $1 \to 0$.

h) Instead of using derivatives w.r.t X & Y axis, like standard edge detection, Canny algorithm uses the directional derivative as the change is more in the direction perpendicular to the vector/edge.

And, instead of detecting edge in the transitions, the Canny algorithm detects edges as the maximum value in the direction of $\nabla(I * G)$

Condition to detect an edge in Canny:

if $\nabla(I * G) >$ a threshold z, detect edge at max in direction of $\nabla(I * G)$

i) Maximum supression in Canny edge detection algorithm is achieved by only taking ~~the~~ or detecting edge in the maximum in the direction of $\nabla(I * u)$ and ignoring all others value or pixels.

Hysteresis thresholding in Canny algorithm. Works by using 2 thresholds, namely Z_H & Z_L

- Start detecting edges with Z_H
- Once an edge is detected, continue tracking on the edge using Z_L.

This part of the algorithm results in finding more connected contour in the image, as this may find out some pixels which are low intensity but forms an edge by ~~using~~ the help of any of the neighboring high intensity pixel.