04/01/2023

NAME : NDIAYE AICHA
· SURNAME : ABDERAHIM
ID_NUMBER : 2010213522

# HOMEWORK

1. Difference between Normalization and standardization

=> Normalization is about involving scaling pixels values to a specific range, often between 0 and 1. standardization, on other hand is about transforming pixel values to have a mean of 0 and a standard deviation of 1. so we should know also that both of them are to enhance image data for better analysis.

2. Bit plane slicing

=> Let's say we have the following image that we will apply Bit plane slicing with it.

| | | | |
|---|---|---|---|
| 4 | 3 | 5 | 2 |
| 3 | 6 | 4 | 6 |
| 2 | 2 | 6 | 5 |
| 7 | 6 | 4 | 1 |

our interval is [0-7] so L=8 and n=3

→ First we have to convert our input image into 3 bit image. suppose we have :

0 - 000
1 - 001
2 - 010
3 - 011
4 - 100
5 - 101
6 - 110
7 - 111

After converting into binary we will convert it into 3-bit image

| | | | |
|---|---|---|---|
| 100 | 011 | 101 | 010 |
| 011 | 110 | 100 | 110 |
| 010 | 010 | 110 | 101 |
| 111 | 110 | 100 | 001 |

3. Histogram Equalization

=> Let's perform the histogram equalization for an 8X8 image shown below

$f(x,y)=$

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 1 |
| 2 | 5 | 3 | 5 | 2 |
| 2 | 5 | 5 | 5 | 2 |
| 2 | 5 | 3 | 5 | 2 |
| 1 | 1 | 1 | 2 | 1 |

input image

max value = 5

Actually here we don't have the number of pixels so let's find it first

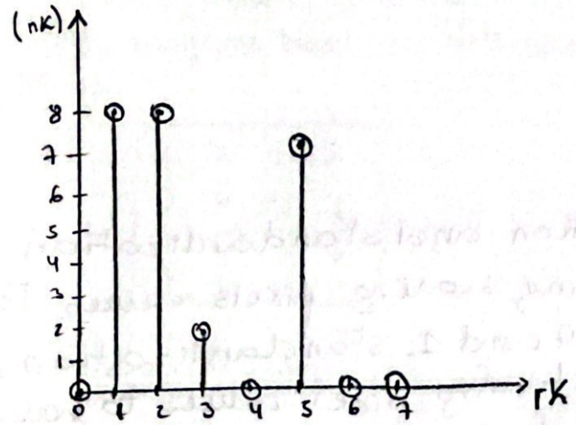
we will write 5 as a power of 2.

$2^2=4$, $2^3=8$ } so $5<8$ => $2^3=8$ => $L=8$

so since interval is $[0, L-1]$ => $[0, 7]$

| Gray levels ($r_k$) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| N° of pixels ($n_k$) | 0 | 8 | 8 | 2 | 0 | 7 | 0 | 0 |

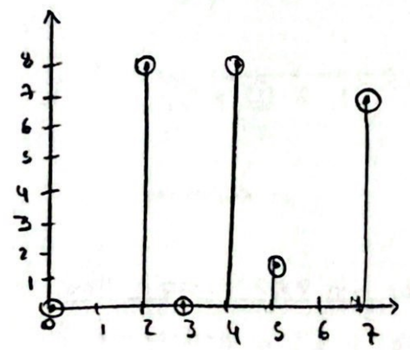
→ So now let's draw the histogram of the input image.




| Gray levels (rK) | Nº of pixels (nK) | $p(rK) = nk/n$ (PDF) | SK CDF | SK x 7 | Histogram Equalization |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | → 0 |
| 1 | 8 | 0,32 | 0,32 | 2,24 | → 2 |
| 2 | 8 | 0,32 | 0,64 | 4,48 | → 4 |
| 3 | 2 | 0,08 | 0,72 | 5,04 | → 5 |
| 4 | 0 | 0 | 0,72 | 5,04 | → 5 |
| 5 | 7 | 0,28 | 1 | 7 | → 7 |
| 6 | 0 | 0 | 1 | 7 | → 7 |
| 7 | 0 | 0 | 1 | 7 | → 7 |
| | n=25 | | | | |

→ now let's redraw our table.

| Gray levels | 0 | 2 | 4 | 5 | 7 |
|---|---|---|---|---|---|
| Nº of pixels | 0 | 8 | 8 | 2 | 7 |




=

| | | | | |
|---|---|---|---|---|
| 2 | 4 | 2 | 2 | 2 |
| 4 | 7 | 5 | 7 | 4 |
| 4 | 7 | 7 | 7 | 4 |
| 4 | 7 | 5 | 7 | 4 |
| 2 | 2 | 2 | 4 | 2 |

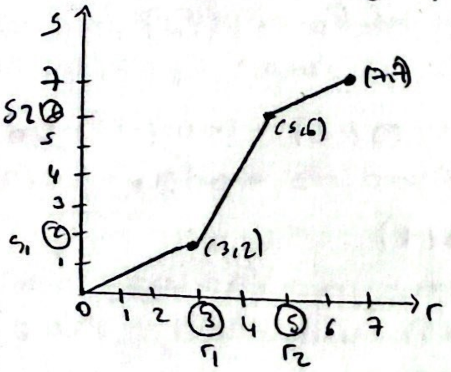
# 4- Contrast stretching

let's say we have the following picture

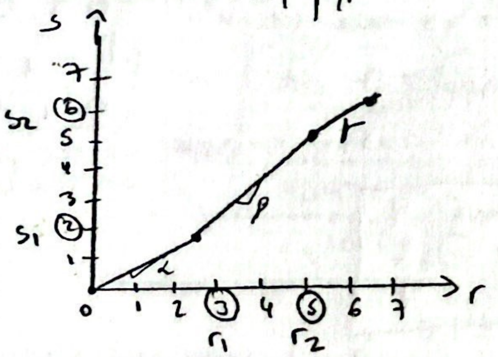
| | | | |
|---|---|---|---|
| 4 | 3 | 5 | 2 |
| 3 | 6 | 4 | 6 |
| 2 | 2 | 6 | 5 |
| 7 | 6 | 4 | 1 |

→ let's say we have $r_1 = 3$ $r_2 = 5$
$s_1 = 2$ $s_2 = 6$

→ we will draw the graph




Now we have to calculate the loop of each length that is why we will give them names as: $\alpha$, $\beta$, $r$




the loop formula is:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\Rightarrow \alpha = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2-0}{3-0} = \frac{2}{3} = 0.66$$

$$\Rightarrow \beta = \frac{y_2 - y_1}{x_2 - x_1} = \frac{6-2}{5-3} = \frac{4}{2} = 2$$

$$\Rightarrow \gamma = \frac{y_2 - y_1}{x_2 - x_1} = \frac{7-6}{7-5} = \frac{1}{2} = 0.5$$

→ so based on these values on $r_1$ and $r_2$, we will get the following functions

$$s = \begin{cases} \alpha . r & 0 \le r < 3 \quad (0,1,2) \\ \beta (r - r_1) + s_1 & 3 \le r < 5 \quad (3,4) \\ \gamma (r - r_2) + s_2 & 5 \le r \le 7 \quad (5,6,7) \end{cases}$$

| r | s |
|---|---|
| 0 | $s = \alpha . r = 0.66 \times 0 = 0$ |
| 1 | $s = \alpha . r = 0.66 \times 1 = 0.66$ |
| 2 | $s = \alpha . r = 0.66 \times 2 = 1.32$ |
| 3 | |
| 4 | $s = \beta (r - r_1) + s_1 = 2(3-3) + 2 = 2$ |
| 5 | $s = \gamma (r - r_2) + s_2 = 0.5(6-5) + 6 = 6$ |
| 6 | $s = \gamma (r - r_2) + s_2 = 0.6(6-5) + 6 = 6$ |
| 7 | $s = \gamma (r - r_2) + s_2 = 0.5(7-5) + 6 = 7$ |

we know that if we have a value less than 0.5 we consider as more than as 0.5

so $0.66 \simeq 1$
$1.32 \simeq 1$
$6.5 \simeq 7$

so now by looking at our input image, we are going to make the output image by considering the values that we calculated.

| | | | |
|---|---|---|---|
| 4 | 2 | 6 | 1 |
| 2 | 7 | 4 | 7 |
| 1 | 1 | 7 | 6 |
| 7 | 7 | 4 | 1 |

output image

## 6) Average Filter

→ let's apply filter to the pixel indicated by the * using a 3x3 neighborhood (matrix)

| | | | |
|---|---|---|---|
| 3 | 9 | 11 | 2 |
| 7 | 15* | 8 | 8 |
| 10 | 12 | 9 | 10 |
| 1 | 9 | 11 | 2 |

New pixel: (3+9+11+7+15+8+10+12+9)/9
= 9, so it will be replacing the "15".

→ now let's see the weighted Average filter.

Here is our mask:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

New pixel: (3(1)+9(2)+11(1)+7(2)+15(4)+8(2)+10(1)+12(2)+9(1))/16
=16, so it will replace the "15"

## 7. Gaussian filter.

→ considering the following image, let's apply our mask on it.

| | | | |
|---|---|---|---|
| 3 | 9 | 11 | 2 |
| 7 | 15 | 8 | 8 |
| 10 | 12 | 9 | 10 |
| 1 | 9 | 11 | 2 |

input image

→ we will apply our mask in the pixel (2,2)

Here is our mask: $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

$\Rightarrow \frac{1}{6}\left[3(1) + 2(9) + 1(11) + 7(2) + 4(15) + 2(8) + 1(10) + 12(2) + 9(1)\right]$

$\Rightarrow \frac{1}{6}(165)$

$\Rightarrow$ 28, it will replace the pixel (2,2).

## 8. Sobel Filter.

$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ this is our mask

the following image is our input image.

| | | | |
|---|---|---|---|
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |
| 50 | 50 | 100 | 100 |

so, now by applying the filter on it on the pixel (1,1), we will have:

$\Rightarrow 50(-1) + 50(-2) + 100(-1) + 50(0) + 50(0) + 100(0) + 50(1) + 50(2) + 100(1)$

$= -50 - 100 - 100 + 50 + 100 + 100$

$= 0$, so it will replace the pixel (1,1).

## 9, Blurring Operation with Laplace

Here is our mask $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

let's apply it on the following image: on the pixel on the center

| | | |
|---|---|---|
| 8 | 5 | 4 |
| 0 | 6 | 2 |
| 1 | 3 | 7 |

$= (8 \times 0) + (5 \times 1) + (4 \times 0) + (0 \times 1) + (6 \times (-4)) + (2 \times 1) + (1 \times 0) + (3 \times 1) + (7 \times 0)$

$= 5 - 24 + 2 + 3$

$= -14$, so we will remplace the center by -14.

from right until you find the cen

# 10- Smoothing linear filters

Q1- Consider the image below and calculate the output of the pixel (2,2) if smoothing is done using 3X3 neighbourhood using all the filters below:

a) Box/mean filter
b) Weighted average filter
c) Median filter
d) Min filter
e) Max filter

$$\begin{matrix} 1 & 8 & 8 & 0 & 7 \\ 4 & 7 & 9 & 5 & 7 \\ 5 & 4 & 6 & 8 & 6 \\ 4 & 2 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{matrix}$$

since it is precised as 3X3 neighbourhood, so we will just use this part I mean the 3X3 around the center (6)

$$\begin{matrix} 1 & 8 & 8 & 0 & 7 \\ 4 & 7 & 9 & 5 & 7 \\ 5 & 4 & 6 & 8 & 6 \\ 4 & 2 & 0 & 1 & 5 \\ 0 & 1 & 0 & 2 & 0 \end{matrix}$$

3X3

## a) Box filter

Let's remember our mask for box filter

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(since all them are 1, so instead of doing $(7\times1)+(9\times1)$, we will just add them

so,

$$O = \frac{1}{9} \times [7+9+5+4+6+8+2+0+1]$$

$$= \frac{1}{9} \times [42] = 4.66 \approx 5$$

so the output image will be the same image, but instead of 6 we will write 5.

1 8 8 0 7
4 7 9 5 7
; 4 5 8 ;

## b) Weighted average filter

the mask was:

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

so,

$$O = \frac{1}{16} \Big[ (7\times1)+(9\times2)+(5\times1)+(4\times2)+(4\times6)+(8\times2)+(2\times1)+(0\times2)+(1\times1) \Big]$$

$$= \frac{1}{16} [81] = 5.0625 \approx \underline{5}$$

so in the final answer, we will have the same matrix or image, but instead of (6) we write 5.

## c) Median filter

we will arrange all of this elements in ascending order

= 0, 1, 2, 4, 5, 6, 7, 8, 9

so, locate the center element

~~0~~, ~~1~~, ~~2~~, ~~4~~, 5, ~~6~~, ~~7~~, ~~8~~, ~~9~~

(so the center is 5)

Median = 5, (it will replace the 6 in the matrix).

## d) Min filter

it is really simple, we calculate the min value (Here we have 0)

= 0

## e) Max filter

We calculate the max value

= 9

## Sharpening spatial filters

what we have wrote in the previous page was about smoothing now we will talk about Sharpening

Sharpening: spatial diffentiation

### Fundation of sharpening filters

1) First order derivative of a one-dimentional function $f(x)$:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

2) Second order derivative of a one-dimentional function $f(x)$:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

### Laplacian filter

the formula is:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{for } f(x)$$

where for $f(x,y)$ =

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

when we substitute them in $\nabla^2 f$ then:

$$\boxed{\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)} \quad \text{for } f(x,y)$$

| | | |
|---|---|---|
| $f(x-1,y-1)$ | $f(x,y-1)$ | $f(x+1,y-1)$ |
| $f(x-1,y)$ | $f(x,y)$ | $f(x+1,y)$ |
| $f(x-1,y+1)$ | $f(x,y+1)$ | $f(x+1,y+1)$ |

input image

now we generate the mask based on the coefficients that we have in the $f(x,y)$

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

here because their coefficient are one I mean $(f(x-1,y)$

because we have $-4f(x,y)$ in the formula and in the input image $f(x,y)$ is in the center so we put -4 in the center.

### * Laplacian mask

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

For all the masks, the sum of a mask is 0.

[redacted]

## Question

**Q1.** Apply Laplacian filter on the given image on the center pixel

| | | |
|---|---|---|
| 8 | 5 | 4 |
| 0 | 6 | 2 |
| 1 | 3 | 7 |

Solution

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Input image

(we can use any mask

$=(\cancel{8\times0})+(5\times1)+(\cancel{4\times0})+(\cancel{0\times1})+(6\times(-4))+(2\times1)+(\cancel{1\times0})+(3\times1)+(\cancel{7\times0})$

$=5-24+2+3$

$=-14$ so we will substitute it instead of 6

## Enhanced Laplacian filter

we add 1

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} 0 & 1 & 0 \\ 1 & -5 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \xrightarrow{\text{Enhanced}} \begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$$

**Q2:** Apply the enhanced Laplacian filter on the given image on the center pixel

| | | |
|---|---|---|
| 8 | 5 | 4 |
| 0 | 6 | 2 |
| 1 | 3 | 7 |

we will use one of the filters

Solution:

$$\begin{bmatrix} 8 & 5 & 4 \\ 0 & 6 & 2 \\ 1 & 3 & 7 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & -9 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$=(8\times1)+(5\times1)+(4\times1)+(\cancel{0\times1})+(6\times(-9))+(2\times1)+(1\times1)+(3\times1)+(7\times1)$

$=8+5+4-54+2+1+3+7$

$=30-54$

$=-24$, we substitute it to the 6

**Q3.** Apply Laplacian filter on the given image

50, 50, 100,
50, 50 (-8) 100,
50, 50, 100,

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 50 | 50 | 50 | 100 | 100 | 100 |
| 1 | 50 | 50 | 50 | 100 | 100 | 100 |
| 2 | 50 | 50 | 50 | 100 | 100 | 100 |
| 3 | 100 | 100 | 100 | 50 | 50 | 50 |
| 4 | 100 | 100 | 100 | 50 | 50 | 50 |
| 5 | 100 | 100 | 100 | 50 | 50 | 50 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | -8 | 1 |
| 1 | 1 | 1 |

mask

Actually we can't directly apply it because they did not tell us in which pixel we should do it or even they did not mention about edges.

So we duplicate the pixels for the edges, then apply the mask

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| 50 | 50 (-8) | 50 | 50 | 100 | 100 | 100 | 100 |
| 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |

$\Rightarrow (50 \times 8) - (50 \times (-8)) = 0$ then write it in the new input image then move the mask to the next.

$\Rightarrow (50 \times 8) + 50(-8) = 0$

$\Rightarrow (50 \times 5) + (100 \times 3) + (50 \times (-8))$

$= 250 + 300 - 400$

$= 550 - 400 = 150$

we will continue like this until we came to

| | | |
|---|---|---|
| 100 | 100 | 100 |
| 100 | 100 | 100 |
| 100 | 100 | 100 |

also going down starting from

| | | |
|---|---|---|
| 50 | 50 | 50 |
| 50 | 50 | 50 |
| 150 | 150 | 50 |

we will keep doing it until finding all the pixels.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 150 | -150 | 0 | 0 |
| 0 | 0 | 150 | -150 | 0 | 0 |
| 150 | 150 | 200 | -200 | -150 | -150 |
| -150 | -150 | -200 | 200 | 150 | 150 |
| 0 | 0 | -150 | 150 | 0 | 0 |
| 0 | 0 | -150 | 150 | 0 | 0 |

this will be our answer

50, 100, 100,
50, 100 (-8) 100,
50, 100, 100,

$\Rightarrow (50 \times 3) + (100 \times 5) + (100 (-8))$

$= 150 + 500 - 800 = 650 - 800$

15. Opening and closing operations

let's say we have segment A and segment B

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 0 \\ 1 & \boxed{1} & 1 \\ 0 & 1 & 0 \end{matrix}$$

segment A $\qquad$ segment B

→ how to find opening $(A \circ B)$

$\Rightarrow A \circ B = (A \ominus B) \oplus B$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$= \boxed{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}}$$

→ how to find closing $(A \bullet B)$

$\Rightarrow A \bullet B = (A \oplus B) \ominus B$

$$= \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \ominus \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

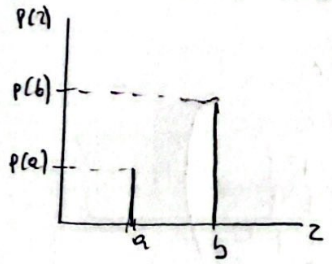
## 12- Salt and Pepper noise

It occurs mostly because sensor and memory problem because of which pixels are assigned incorrect maximum values

the pdf of impulse Noise is given by:

$$p(z) = \begin{cases} P_a & \text{for } z=a \\ P_b & \text{for } z=b \\ 0 & \text{otherwise} \end{cases}$$

If either $P_a$ or $P_b$ is zero, impulse noise is called unipolar




## 13. Contraharmonic mean filter

→ It is well suited for reducing effects of salt and pepper noise.

→ Q > 0 for elimination of pepper noise and Q < 0 for elimination of salt noise

→ Q is the order of filter

$$f(x,y) = \frac{\sum g(s,t)^{Q+1}}{\sum g(s,t)^{Q}}$$

where Q = 0, it becomes arithmetic mean filter

where Q = 1, it becomes harmonic mean filter.

• Mean filter.

Here is our mask: $\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

lets apply it on the following image:

| | | |
|---|---|---|
| 7 | 9 | 5 |
| 4 | 6 | 8 |
| 2 | 0 | 1 |

we will apply the mask on the pixel on the center

$= \frac{1}{9} \times [7+9+5+4+6+8+2+0+1]$

$= 42/9$

$= 5$, that will replace 6

## 14. RGB color model

suppose we have an image

so,

→ each color = Red, Green & blue

| | |
|---|---|
| (x, y, z) R G B | black (0,0,0) |
| R (1,0,0) | white (1,1,1) |
| G (0,1,0) | yellow = R+G |
| B (0,0,1) | yellow (1,1,0) |

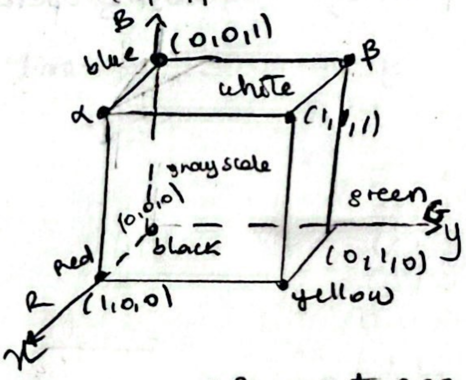
So based on our image, the rightside that we call $\alpha = G+B$

$\alpha = (0,1,1)$

and the left side that we call

$\beta = R+B$

$= (1,0,1)$




Red → 8 bit → 0 to 255

Green → 8 bit → 0 to 255

Blue → 8 bit → 0 to 255

pixel depth = 24 bit.

A 24 bit color image = full color image.

The total number of color in 24 bit RGB image is

$2^{24} = 16777216$.