~~Image~~

Image -

An image is defined as a two- dimensional func-tion, $F(x,y)$, where x and y are spatial coordinates, and the amplitude of F at any pair of coordinates (x,y) is called the intensity of that image at the point. When x,y and amplitude valuees of F are finit we call it digital image.

In other words, an image can be defined by a two- dimensional array specifically arranged in rows and columns.

Digital Image processing →

Digital Image processing (DIP) is a software which is used to manipulate the digital images by the use of computer system. It is also used to enhance the images, to get some important information from it.

Forexample, Adobe photoshop, MATLAB etc.

It is used in the conversion of signals from an image sensor into the digital images.

- A certain number of algorithms are used in image processing.
- It provides images in different formats.
- DIP provides a platform to perform various operation like image enhancing, processing of analog and digital signals, image signals, voice signals etc.

Digital Image Processing allows users the following tasks →

- Image Sharpening and restoration : The common application of image sharpening and restoration are zooming, blurring, sharpening, grayscale conversion, edge detection, Image recognition, and image retrieval, etc.

- Media field : The common applications of media field are Gamma-ray imaging, PET scan, X-Ray imaging, Medical CT, UV imaging, etc.

- Remote sensing : It is the process of scanning the earth by the use of satellite and acknowledges all activities of space.

Machine/ Robot vision: It works on the vision of robots so that they can see things, identify them etc.

Characteristics of Digital Image processing:-

- It uses software, and some are free of cost.
- It provides clear images.
- Digital Image processing do image enhancement to recollect the data through images in many fields,
- It is used widely everywhere
- It reduce the complexity of dIP
- It is used to support a better experience of life.

→ Advantages of DIP-

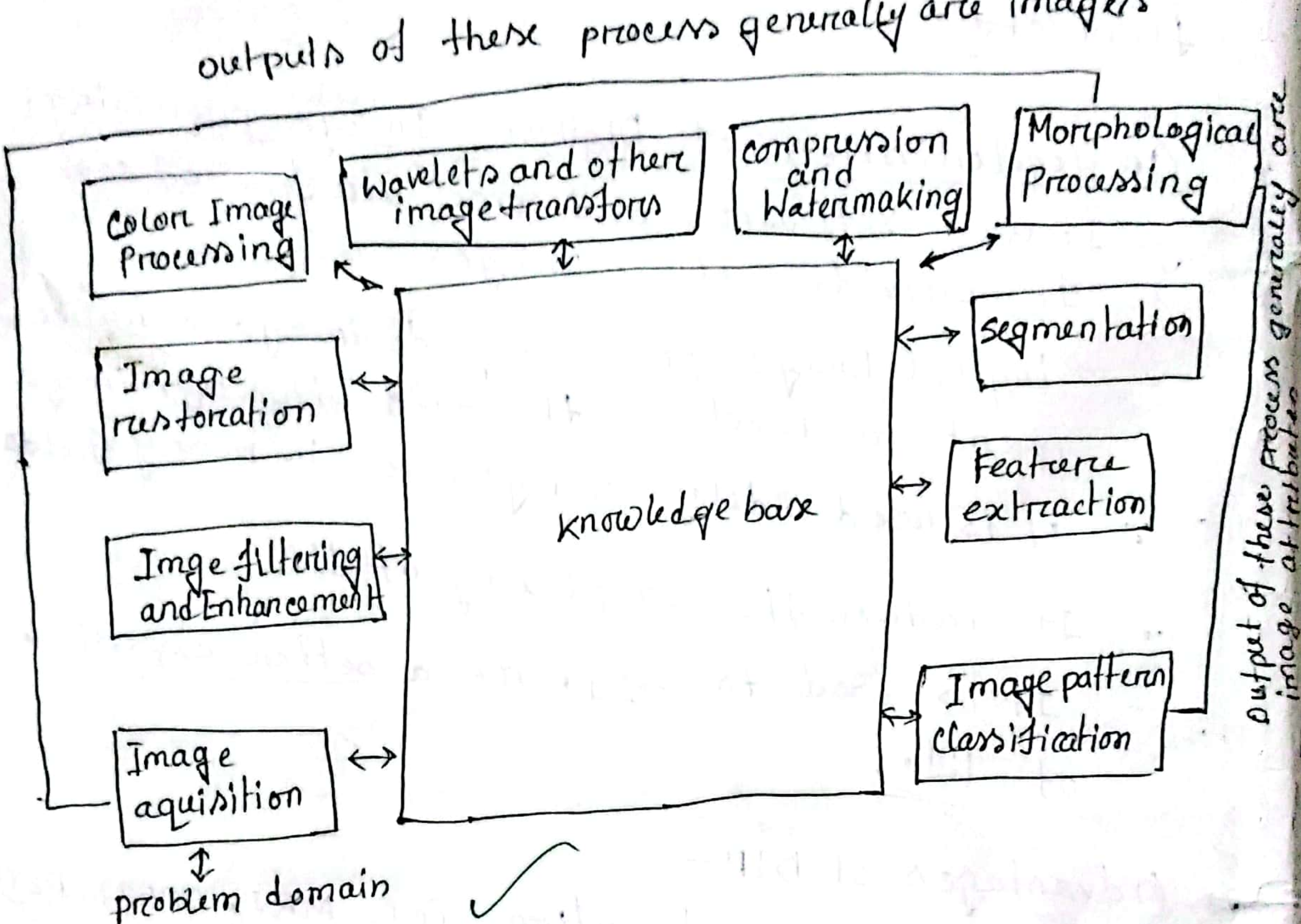
- Image reconstruction (CT, MRI, SPECT, PET)
- Image reformatting (Multi-plane, multi-view reconstructions)
- First image storage and retrieval.
- First and high quality image distribution
- controlled viewing (windowing, zooming)

→ Disadvantages:

→ It is very much time consuming

→ It is very much costly depending on the particular system

→ Qualified persons can be used.



- **Image aquisition** : It is the first process. Acquisition could be as simple as being given an image that is already in digital form. Generally, the image aquisition stage involve preprocessing such as (scaling).

- **Image Enhancement** : It is the process of manipulating an image so that result is more suitable than the original for a specific application. This can involve techniques such as histogram equalization, contrast stretching and sharpening.

Image Restoration: is an area that ~~???~~ also deals with improving the appearance চেহারা of an image. This is particularly useful in scenarios where images are affected by blurring অস্পষ্ট, motion গতি or other forms of distortion বিকৃতিগুলো.

Color image processing: is an area that has been gaining importance because of the significant increase in the use of digital images over the internet. Colour is used also as the basis for extracting feature of interest in an image.

Wavelets: are the fundation for representing images in various degree of resolution.

Compression: Reducing the storage space required to store images space required to store images while minimizing loss of image quality. Compression techniques: include lossless methods and lossy methods.

Morphological : processing deals with tools for extracting image components that are useful in the representation and description of shape.

Image Segmentation: Partitioning an image into meaningful regions or objects to facilitate further analysis. Segmentation is crucial for tasks such as object detection recognition, and tracking.

Feature Extraction: Identifying and quantifying relevant patterns or features within the image such as edges, corners, textures, or shapes. Feature extraction is essential for tasks like object recognition and classification.

Image pattern classification: is the process that assigns a label (e.g., "vehicle") to an object based on its feature description.

Components of an Image processing system:→

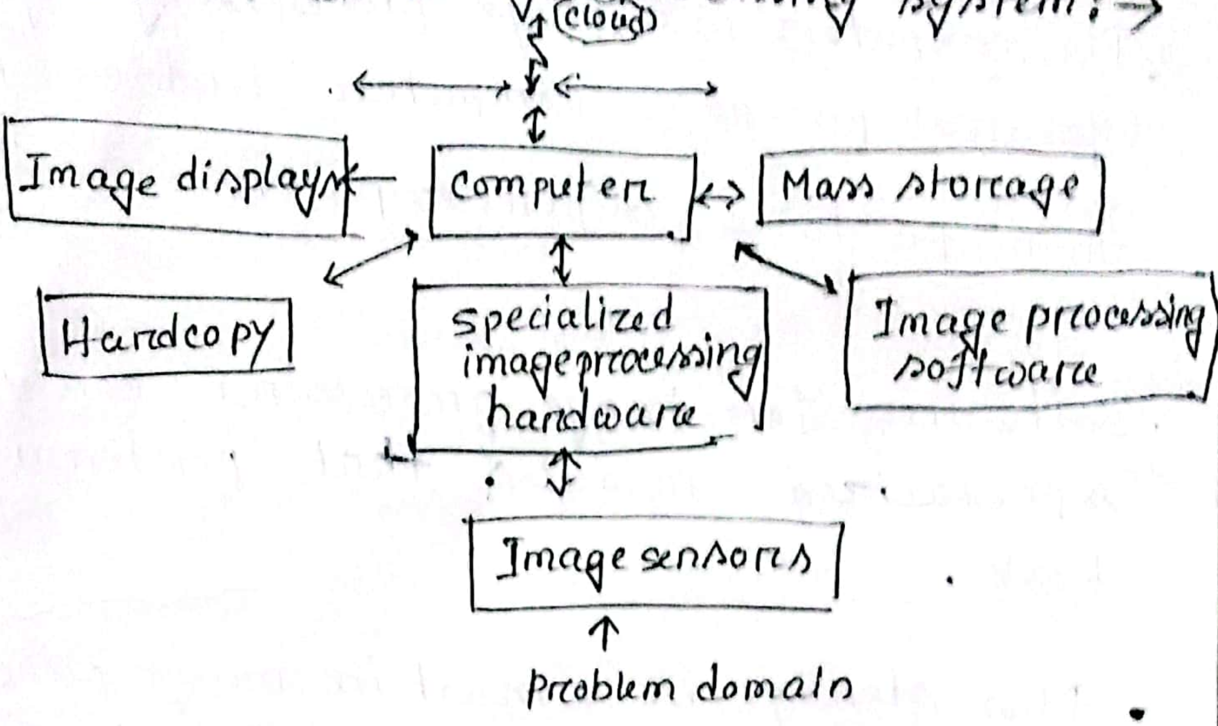


Fig: Components of general purpose of Image processing system.

The subsystems are required to acquire digital images. The first is physical sensor that responds to the energy rediated by the object we wish to image. The second call is digitizer. is a device for converting the output of the physical sensing in digital forms.

... Specialized image processing hardware usually consists of the digitizer just mentioned, plus hardware that performs other primitive operations, such as an arithmetic logic unit (ALU), that performs arithmetic and logical operation in parallel on entire images.

- The computer in a image processing system is a general purpose computer and can range from pc to a supercomputer.

- Software for image processing consists of specialized modules that perform specific task.

- Mass storage is a must in image processing applications. An image of size 1024 x 1024 pixels in which the intensity of each pixel is an 8-bit quantity, requires one megabyte of storage space if the image is not compressed.

Image displays is used for mainly color, flat screen monitors. Monitors are driven by the outputs of image and graphics display cards that are an integral part of the computer system.

Hardcopy devices for recording image include last printers, film cameras, heat sensitive devices, ink-jet units and digital units, such as optical and CD-ROM disks.

Network and cloud communication are almost default function in any computer system in use today. Because of the large amount of data inherent in image processing application, the key consideration image transmission is bandwidth.

"Digital Image Fundamentals"

Gray level: A gray level refers to the intensity value of a pixel in a digital grayscale image. It represents the brightness of the pixel and typically ranges from 0 (black) to 255 (white) in an 8-bit gray scale image. Each pixel in the image is assiged is single gray level value, indicating its luminance or brightness level.

Gray Scale: Gray scale is a type of image in which each pixel is represented by a single intensity value, usually corresponding to its gray level. Unlike color images which contain multiple channels, grayscale images have only one channel respresenting brightness.

grayscale images are commonly used in applications where color information is not necessary such as medical imaging, document processing. They are simpler to process and required less storage space compared to color images.

How much memory needed to store a gray image?

To determine how much memory is need to store a grayscale image. We need to consider the following factors:→

1. Image Dimensions : The width and height of the image in pixels.
2. Bit Depth : The number of bits used to represent each pixel's intensity. Common choice are 8 bits (1 byte) per pixel for 256 shades of gray or 16 bits (2 bytes) per pixel for higher precision.
3. Compression: Whether the image is compressed or stored in a raw, uncompressed format.

Assuming an 8-bit grayscale image (256 shades of gray) without compression: →

- Image Size in Bytes:

  Image width x Image height x Bytes per pixel

- Total Memory needed: Image size in Bytes + Additional overhead

For ex, if ~~you~~ we have a grayscale image with dimensions 1024 x 768 pixels →

Image size in Bytes = 1024 pixels (width) x 768 pixels (height) x 1 byte/pixel

= 786,432 bytes.

So, the total memory needed to store this grayscale image would be approximately 786,432 bytes, or roughly 768 klibytes (KB)

A Simple image formation model: →

An images ~~by~~ is represented by two-dimensional functions of the form f(x,y). The value of f at spatial co-ordinates (x,y) is a scalar quantity whose physical meaning is determined by the source of the image, and whose values are proportional to energy radiated by a physical source.

As a consequence $f(x,y)$ must be non-negative and finite, that is,

$$0 \leq f(x,y) < \infty$$

Function $f(x,y)$ is characterized by two components

1. the amount of source illumination incident on the scene being viewed and
2. the amount of illumination reflected by the objects in the scene.

Appropriately, these are called the illumination and reflectance components and are denoted by $i(x,y)$ and $r(x,y)$ respectively. The two functions combine as a product to form $f(x,y)$.

$$f(x,y) = i(x,y)\, r(x,y)$$

where $\quad 0 \leq i(x,y) < \infty$

and $\quad 0 \leq r(x,y) \leq 1.$

Thus, reflectance is bounded by 0 (total absorption) and 1 (total reflectance). The nature of $i(x,y)$ is determined by the illumination source, and $r(x,y)$ is determined by the characteristics of the imaged object.

These expressions are applicable also to images formed via transmission of the illumination through a medium, such as a chest X-ray.

Image sampling and Quantization :→

There are numerous ways to accquire images, but our objective in all is the same: to generate digital images from sensed data. The output of most sensors in a continuous valtage waveform, whose amplitud and spatial behavior are related to the physical phenomenon being sensed. To create a digital image, we need to convert the continuous sensed data into a digital format. This requires two proccesses.→

1. Sampling and 2. Quantization.

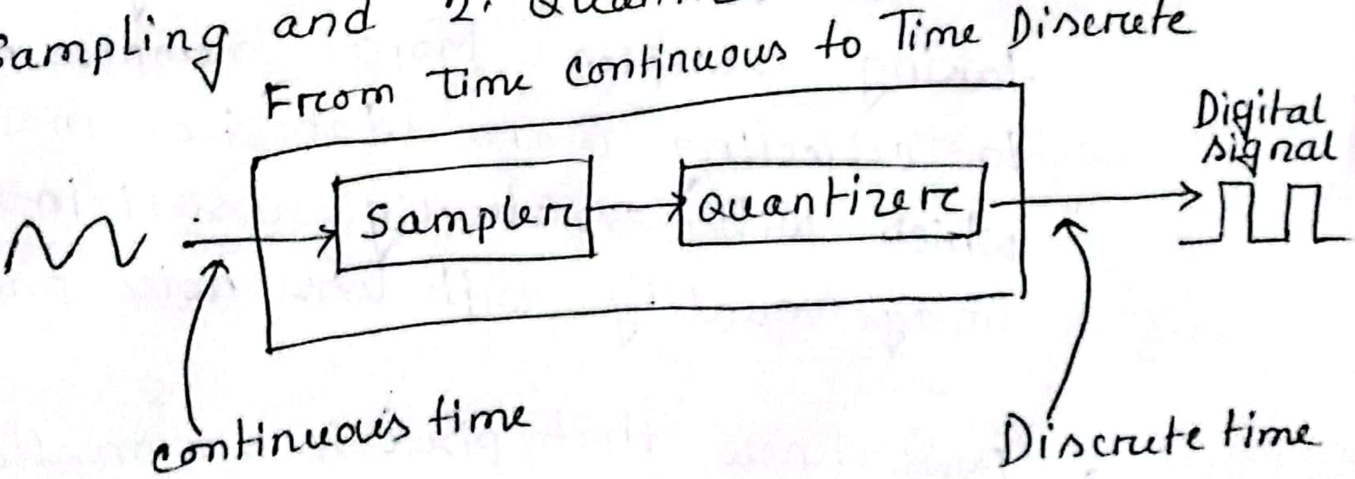
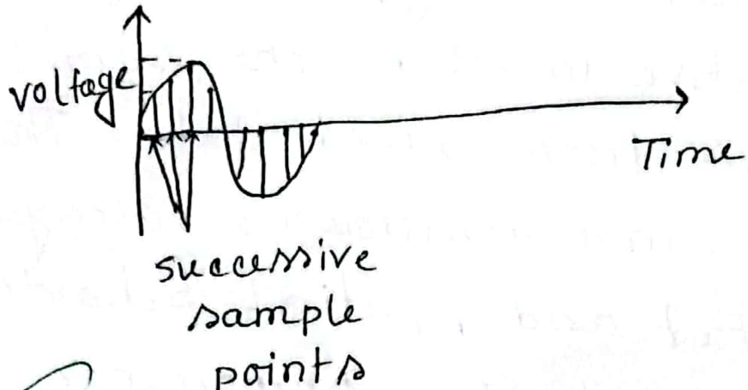


Fig: Analog to Digital conversion.

Sampling in Digital Image processing:

- In this we digitize x-axis in sampling
- It is done on the independent variable
- For e.g. if $y = \sin x$, it is done on x variable.



- There are some variation in the sampled signal which are random in nature. These variations are due to noise.

- We can reduce this noise by making taking samples. More samples refer to collecting more data i.e. more pixels which will eventually result in better image quality with less noise present.

- As we know that pixel is the smallest element in an image and for an image represented in the form of a matrix total no. of pixels is given by;

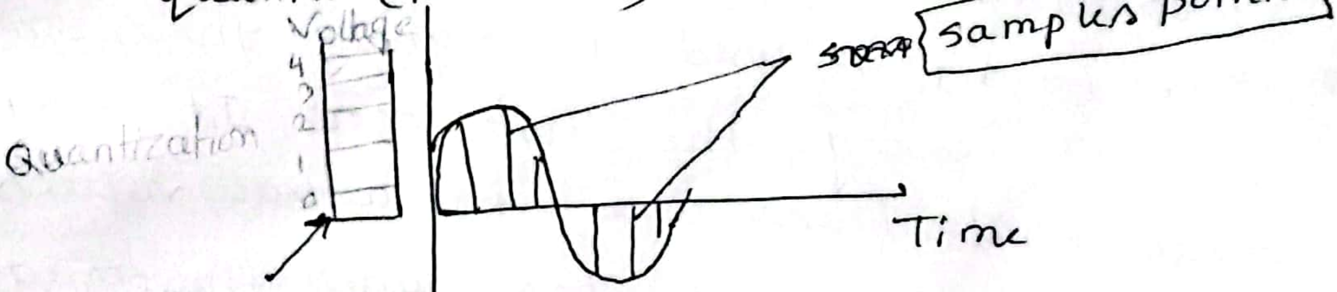
Total number of pixels = Total number of rows × Total number of columns.

- The number of samples taken on the x-axis of a continuous signal refers to the number of pixels of that image.

Quantization in Digital Image Processing →

- It is the opposite of sampling as sampling is done on the x-axis, while quantization is done on the y-axis.

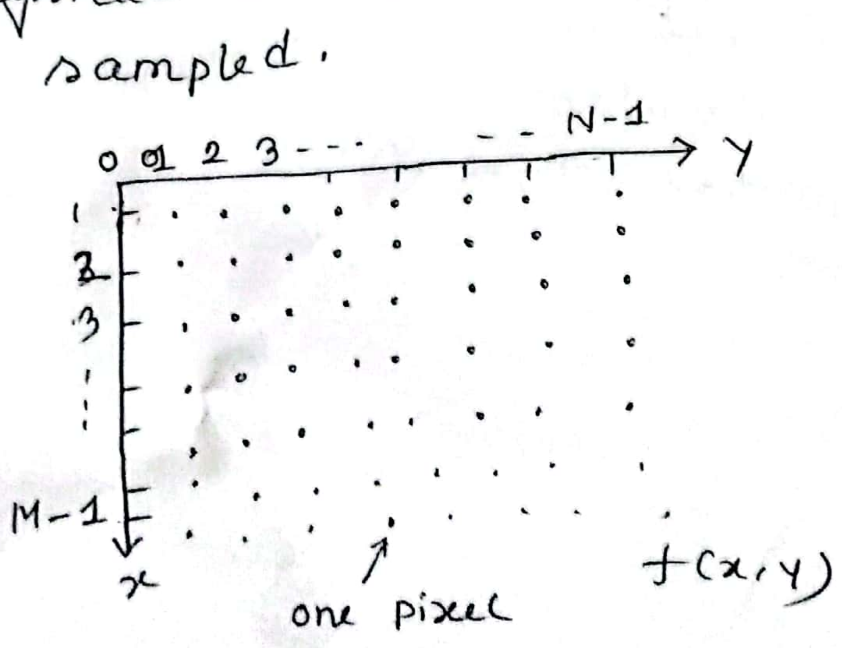
- Digitizing the amplitude is quantization. In this we divide the signal amplitude into quanta (partitions)



vertically ranging values have been quantize into 5 different levels or partitions

Representing Digital Images:

We will use two principle ways to represent digital images. Assume that an image $f(x,y)$ is sampled so that the resulting digital image has M rows and N columns. The values of the coordinates (x,y) now become discrete quantities. For notation clarity and convenience, we shall use integer values for these discrete co-ordinates. Thus the values of the co-ordinates at the origin are $(x,y) = (0,0)$, The next co-ordinate values along the first row of the image are represented as $(x,y) = (0,1)$. It is important to keep in mind that the notation $(0,1)$ is used to signify the second sample along the first row. It does not mean that these are the actual values of physical co-ordinates when the image was sampled.



The notation introduced in the preceding paragraph allows us to write the complete M×N digital image in the following compact →

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & & & \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Each element of this matrix array is called an image element.

Some basic Relationships between pixels-

Neighbors of pixel:

A pixel p at co-ordinates (x,y) has two horizontal and two vertical neighbors with coordinates

$$(x+1,y),\ (x-1,y),\ (x,y+1),\ (x,y-1)$$

This set of pixels called the 4-neighbors of pixel p, is denoted $N_4(P)$

The four diagonal neighbors of p have coordinates →

$$(x+1,y+1),\ (x+1,y-1),\ (x-1,y+1),\ (x-1,y+1)$$

and are denoted $N_D(p)$. These neighbors, together with the 4-neighbors are called the 8-neighbors of p, denoted by $N_8(p)$.
The set of image locations of the neighbors of a point (p) is called the neighborhood of p. The neighborhood is said to be closed if it contains p. otherwise, the neighborhood is said to be open.

connectivity:

Connectivity between pixels is a fundamental concept that simplifies the definition of numerous digital image concepts, such as regions and boundaries. To establish if two pixels are connected, it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity. For instance, in a binary image with values 0 and 1, two pixels may be 4-neighbors but they are said to be connected if they have the same value.

adjacency

Let V be the set of a gray level values used to define adjacency. In a binary image, $V = \{1\}$ if we are referring to adjacency of pixels with value 1. In a grayscale image the idea is the same but set V typically contains more elements. For example, in the adjacency of pixels with a range of possible gray-level values 0 to 255, set V could be any subset of these 256 values.

We consider three types of adjacency →

1. 4- adjacency. Two pixels p and q with values from V are 4- adjacency if q is in the set $N_4(P)$

2. 8- adjacency : Two pixels p and q with values from V are 8- adjacent if q is in the set $N_8(P)$

3. m- adjacency. Two pixels p and q with values from V are m adjacent if

(i) q is in $N_4(P)$ or

(ii) q is in $N_D(P)$ and the set has no $N_4(P)$ or pixels whose values are from V.

A path from pixel (P) with co-ordinates (x,y) to pixel q with co-ordinates (s,t) is a sequence of distinct pixels with coordinates where and pixels are adjacent for. In these case, n is the length of the path. If $(x_0, y_0) = (x_n, y_n)$, the path is a closed path. We can define 4-, 8-, or m-path depending on the type of adjacency specified.

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

1. $D(p,q) \geq 0$, $D(p,q) = 0$ if $p = q$
2. $D(p,q) = D(q,p)$
3. $D(p,z) \leq D(q,z) + D(p,q)$

$D(p,q) \geq 0$, $D(p,q) = 0$ if $p = q$

$D(p,q) = D(q,p)$

$D(p,z) \leq D(q,z) + D$

Distance Measures

Given pixels p, q, and z, with co-ordinates (x, y), (s, t) and (v, w) respectively. The distance function D has following properties or Metric id :

p, q, and z
x, y s, t
v, w

(a) $D(p, q) \geq 0$, $D(p, q) = 0$ if $p = q$

(b) $D(p, q) = D(q, p)$

(c) $D(p, z) \leq D(q, z) + D(p, q)$

$D(p, q) \geq 0$ D
$p(p, q)$
$p(p, z)$

The following are the different distance measures :

1 - The Euclidean distance between p and q is defined as :

$$D_e(p, q) = \{(x-s)^2 + (y-t)^2\}^{1/2}$$

for this distance measure, the pixels having a distance less than or equal to some value from (x, y) are the point contained in a disk of radius r centered at (x, y).

2 - The city-Block Distance

$$D_4(p, q) = |x-s| + |y-t|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

3 - The chessboard Distance

$$D_8(p, q) = \max(|x-s|, |y-t|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Q: A image segment is shown below. Let v be the set of gray level value used to define connectivity in the image. Compute D4, D8 and DM distance between pixels 'p' and 'q' for:

(i) $V = \{2, 3\}$

ii. $V = \{2, 6\}$

2(p)	3	2	6	1
6	2	3	6	2
5	3	2	3	5
2	4	3	5	2
4	5	2	3	6(q)

$D4(p,q) = |x-s| + |y-t|$

$= |0-4| + |0-4|$

$= 4+4 = 8$

$D8(p,q) = \max(|x-s|, |y-t|)$

$= \max(|0-4|, |0-4|)$

$= \max(4, 4)$

$= 4$

co-ordinate of $p(x,y) = (0,0)$

co-ordinate of $q(s,t) = (4,4)$

(i) $V = \{2, 3\}$ There is no path between p and q as q(6) is not included in the set v.

(ii) $V = \{2, 6\}$ There is no path between p and q.

Arithmetic operation between image: →

There are array operation which are carried out between corresponding pixel pairs. The four arithmetic operation are denoted as:

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

s (m... fin
d
p
v

Four important points: →

→ If the result is floating point number round off its value.

→ If the result is above the pixel range, select the max rang value.

→ If the result is below the pixel range select the min range value.

→ If the result is infinite, write is as zero.

Addition:

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} + \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 10 & 200 & 15 \\ 6 & 0 & 10 \\ 8 & 10 & 15 \end{bmatrix}$$

Uses: → Addition of noisy images for noise reduction.

→ Image averaging in the field of astronomy.

subtraction:

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} - \overset{B}{\begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix}} = \begin{bmatrix} 0 & 0 & 5 \\ 2 & 0 & 10 \\ 8 & 0 & 0 \end{bmatrix}$$

uses:

- Enhancement of difference between images.
- Mask mode radiograph in medical imaging.

Multiplication:

$$\overset{A}{\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix}} \times \begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 255 & 50 \\ 8 & 0 & 0 \\ 0 & 0 & 50 \end{bmatrix}$$

uses:

- shading correction.
- Masking or region of interest operations

Division:

$$\begin{bmatrix} 0 & 100 & 10 \\ 4 & 0 & 10 \\ 8 & 0 & 5 \end{bmatrix} \div \begin{bmatrix} 10 & 100 & 5 \\ 2 & 0 & 0 \\ 0 & 10 & 10 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

uses –

→ shading correction.

Image compression:

Image compression:

Image compression refers to the process of redundancy amount of data required to represent the given quantity of information for digital image. The basis of reduction process is removal of redundant data.

In terms of storage, the capacity of a storage device can be effectively increased with methods that compress a body of data on its way to a storage device and decompresses it when it is retrieved.

image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level

Compression Methods:

- Run length Encoding (RLE)
- Arithmetic coding
- Huffman coding
- Transform coding

There are two types of compression →
- Lossy image compression
- Lossless image compression.

Lossy compression: It is a type of data encoding and compression that intentionally discards some data during the compression process.
It reduce the ~~file~~ file size to large extent. Quality of the image also degraded but keep the appearance of the image as interact as possible while making the simplification.

Here we have to compromize with the quality of image. It restores the large file to its original form with loss of some data which can be considered as not-noticible.

Lossless image compression:
→ It is the best method of compression because it give good compression ratio's with minimal loss of image quality.

→ It maintain the quality of image without losing its ~~from~~ form so that's why the compression file size not smaller as compare to lossy compressed file.

→ If we need to decompressed the file in its original form it performed decompression without any loss of data and restore the large file to its original form.

→ It keeps even those bytes which are not noticeable.

Need for data compression :→

→ If data can be effectively compressed where possible, significant importants in data throughout can be achieved.

→ Compression can reduce the file sizes upto 60-70 % and hence many files can be combined into compression document which makes the sending easier.

→ It is needed because it helps to reduce the compress consumption of excessive resource such as hard disk space and transmission bandwidth.

→ Compression can fit more data in small memory and thus it reduces the memory space required as well as the cost of managing data.

Lossless compression	Lossy compression
1. It does not eliminate the data which is not noticable	1. It eliminate the data which is not noticable
2. File can be restored in its original form	2. File doesn't restore in its original form
3. Quality does not compressed	3. Data quality is compressed.
4. It does not reduce the size	4. It reduce the size.
5. Ex: Run length coding, Huffman coding	5. Ex: Transform coding discrete cosine, wavelet!
6. It is used in Text, images, sound	6. It is used in images, audi, vedio.

Data Redundancy:

Relative data redundency:

Let, there are two sets represent the same information and n1 and n2 are the number of information carrying units in the data-sets:

The relative data Redundancy R_d is given that,

$$R_d = (1 - 1/C_R)$$

where, C_R = compression ration $= \frac{n1}{n2}$

i) If $n2 = n1$, no redundant data is present in first set (image input)

ii) If $n2 << n1$, highly redundant data is present in first set.

iii) If $n2 >> n1$, highly redundant data is present is second set.

Types of redundancy:

1. Coding redundancy;
2. Spatial or temporal redundany
3. Psychovisual redundancy (irrelevant Information)

Data compression attempts to reduce one or more of these redundancy types.

codding redundancy: A code is system of symbols used for representing information. A code word is a sequence of symbol us representing a piece of information. Thus the code length is defined as the number of symbol in each code word. A resulting image is said to have coding redundancy if its gray levels are coded using more coded symbols than actually needed to represent each gray level.

Spatial or temporal redundancy →

~~Iat~~ It's caused by the interpixel correlations within an image. Interpixel correlations are the structural and geometric relationships between objects in the image.

ii. psychovisual redundancy:

Most images contain information that is ignored by the human visual system and /or irrelevant to the intended use of the image. It is redundant in the sense that it is not used.

• Psychovisual

Coding Redundancy

- **Code**: a list of symbols (letters, numbers, bits etc)
- **Code word**: a sequence of symbols used to represent some information.

Code word length: number of symbols in a code word.

Ex: (binary code, symbols: 0, 1, length: 3

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

0,1

0-255 for gray

$l(r_k)$

0-255

$r_k \in$ 0-255 for gray scale

- $r_k \rightarrow$ Input Intensity Value e.g. 0-255 for gray-scale image
- $l(r_k) \rightarrow$ No. of bits used to represent r_k.
- The average no of bits required to represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)\, P_r(r_k)$$

say, uniform width str. (all pixel are using 8 bits to represent intensity)

$\rightarrow$ equal length code (fixed length code)

$\rightarrow$ m-bit fixed length code (8 bit fixed length code in our case)

r_k	$P_r(r_k)$	code 1	$L_1(r_k)$	code 2	$L_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	–	8	–	0

Code1 → equal length code

code 2 → variable length code (obt by Huffman coding)

Min bits → higher Probability

Max bits → Lower Probability

- In this case, $L(r_k) = 8$ [each req. 8 bits in fixed length]

$$L_{avg} = 8 \sum_{k=0}^{L-1} P_r(r_k) = 8 * 1 = 8$$

- coding redundancy tries to reduce L_{avg}
- Thus, total no. of bits required to represent an $M \times N$ image is MNL_{avg}
- The code can be an equal length code or variable length code.

For code 1, $L_{avg} = 8$

on the other hand, using code 2, the average length of the encoded pixels is

$$L_{avg} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3)$$
$$= 1.81 \text{ bits}$$

The resulting compression and corresponding relative redundancy are

$$C = \frac{256 \times 256 \times 8}{256 \times 256 \times 1.81} \approx 4.42$$

$$R = 1 - \frac{1}{4.42} = 0.774$$

Thus 77.4 % of the data in the original 8 bits 2-D intensity array is redundant

- Coding redundancy is present when the codes assigned to intensity values do not take full advantage of their probabilities of occurrance.

- The natural consquence is that for most images, certain intensities are more probably than others.

- A natural binary encoding assigns the same number of bits to both the most and least probably values, failing to minimize and resulting in coding redundancy.

Huffman coding

- A measure to reduce coding redundancy
- Most popular coding redundancy technique
- Variable length code
- Min length code is assigned to one with highest probability.

Say, Image size : 10x10 (5 bit image)

Frequency :

$a_2 = 40$ $a_6 = 30$ $a_1 = 10$ $a_4 = 10$ $a_3 = 6$ $a_5 =$

$p(a_2) = 40/100 = 0.4$

Symbols (Like intensity levels)	probabilities (Sorted)	Source Reduction (do till two values are left) (Maintain in sorted order here as well)			
		1	2	3	4
a_2	0.4	0.4	0.4		
a_6	0.3	0.3	0.3		
a_1	0.1	0.1	0.2		
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Symbols (Like intensity levels)	Probabilities (sorted)		Source Reduction (do till two values are left) 1	2	3	4
a2	0.4	1	0.4	0.4	0.4	0.6 (0)
a6	0.3	00	0.3	0.3	0.3 (00)	
a1	0.1	011	0.1	0.2 (010)	0.3 (01)	0.4 (1)
a4	0.1	0100	0.1 (0100)	0.1 (011)		
a3	0.06	01010	0.1 (0101)			
a5	0.04	01011				

Encoded string : 010100111100

Decoding : a3 a1 a2 a2 a6

parameters:

1. Average length of code

$L_{avg} = 0.4 * 1 + 0.3 * 2 + 0.1 * 3 + 0.1 * 4 + 0.06 * 5 + 0.04 * 5 = 2.2$ bits/symbol

2. Total no. of bits to be transmitted

$10 * 10 * 2.2 = 220$ bits

3. entropy $= 2.1396$

How much saved $= \dfrac{10 * 10 * 5 - 10 * 10 * 2.2}{10 * 10 * 5}$

$= 0.56 = 56\%$

fidelity :

Fidelity Criteria:

Fidelity: The degree of exactness (Fragment) with which something is copied or reproduced called fidelity.

To determine exactly what information is important, and able to measure image quality we need to define image fidelity criteria.

can be divided into two types →

1. Objective Fidelity
2. Subjective "

*objective fidelity criteria :

When the level of information loss can be expressed as a function of the original or input image and the compressed and subsequently decompressed output image is said to be based on an objective fidelity criteria

Let $f(x,y)$, denote an estimate approximation of $\hat{f}(x,y)$ that result from compressing and subsequently decompression the input. For any value of x and y, the error $e(x,y)$ between $f(x,y)$ and $\hat{f}(x,y)$ can be defined as;

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

so that the total error between the two images is,

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]$$

- Where the images are of size $M \times N$
- The root mean square e_{rms} between the sequence $f(x,y)$ and $\hat{f}(x,y)$ then is the root of the squared error average over the $M \times N$ array or;

$$e_{rms} = \sqrt{\frac{1}{MN} [\hat{f}(x,y) - f(x,y)]^2}$$

2. Subjective Criteria:

- Measuring image quality by the subjective evaluations of a human observer is often more appropriate since most decompressed images are ultimately viewed by human beings. This can be accomplished by showing a decompression image to a viewers and averaging their evaluations. An example of a rating scale is shown in the following table. The evaluations are said to be based on subjective fidelity criteria.

Rating

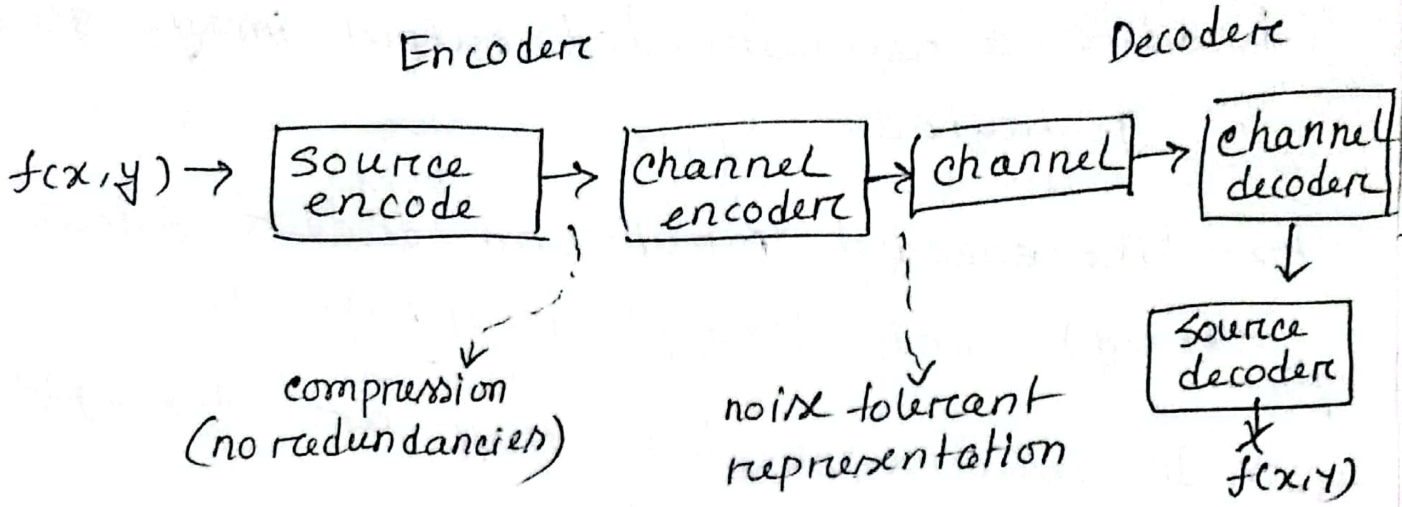
1. Excellent - Very high quality
2. Fine - high quality
3. Passable - Acceptable quality
4. Marginal - poor quality
5. Inferior - very poor image
6. Unsuble - An image so bad that could not watch it.

Image Compression Models →

→ The image compression system is composed of 2 distinct structural blocks: an encoder & a decoder.

→ Encoder performs compression.

→ Decoder performs Decompression.



→ The encoder is made up of a source encoder which removes input redundancies and a channel encoder, which increases the noise immunity of the source encoder's output.

→ The decoder includes a channel decoder followed by a source decoder.

→ If the channel between the encoder and decoder is noise free the channel encoder and decoder are omitted.
বর্জন করা

→ Input image f(x,y) is fed (খাওয়ানো) into the encoder, which creates a compressed representation of input.

→ It is stored for future for later use or transmitted for storage and use at a remote location.

→ When the compressed image is given to

decoder, a reconstructed output image $f'(x,..)$ is generated.

$f'(m)$

→ The encoded input and decoder output are $f(x,y)$ and $f'(x,y)$ respectively.

→ In video application, they are $f(x,y,t)$ & $f'(x,y,t)$ where t is time.

→ ~~This operation~~ is irreversible

1- The source ~~and~~ encoder and decoder →

- The source encoder is responsible for reducing or eliminating any coding, interpixel and psychovisual redundancies in the input image

- Encoder is used to remove the redundancies through a series of 3 independent operations.

- Mapper: It transforms $f(x,y)$ into a format designed to reduce interpixel redundancies.

- It is reversible

- It may / may not reduce the amount of data to represent image.

→ Quantizer: reduces the accuracy of the mapper's output in according with some pre-established fidelity criterion. This stage reduces the psychovisual redundancies of the input image.

→ This operation is irreversible.

→ Encoder:

- Symbol encoder: Generates a fixed or variable length code to represent the quantizer output and maps the output in the accordance with the code.

- In most cases a variable length code is used to represent the mapped and quatized data set. It assigns the shortest code words to the most frequently occurring output values and thus reduces coding redundancy.

- It is reciversible.

- Upon its completion, the input image has been processed for the removal of all 3 redundancies.

$f(x,y) \rightarrow$ Mapper $\rightarrow$ Quantizer $\rightarrow$ Symbol encoder $\rightarrow$ channel

Mapper: no interpixel redundancies (reversible)

Quantizer: no psychovisual redundancies

Symbol encoder: no coding redundancies

Mapper: Transform input data in a way that facilitates reduction of interpixel redundancies.

Quantizer: reduces the accuracy of the mapper's output in accordance with some pre-established fiderlity criteria.

Symbol encoder: assigns the shortest code to the most frequently occurring output values.

- Decoder

Channel $\rightarrow$ symbol decoder $\rightarrow$ De-Quantizer $\rightarrow$ Inverse Mapper $\rightarrow \hat{f}(x,y)$

- Inverse operation are performed
- But -- quantization is irreversible in general
- Quantization results in irreversible loss

an inverse quantizer block is not included in the decoder block

Image Enhancement

"Intensity Transformation and spatial Filtering"

Image Enhancement: It is a process that improves the quality of an image for a specific application.
The reason for the image enhancement to highlight the important details and to remove noise from image and also make image to more appealing.

3 methods that are used for image Enhancement:

1. Spatial domain: Direct manipulate of pixel values.

2. Frequency domain: Modifying the Fourier Transform of image.

3. Combination method: Combination of first and second method.

Spatial Domain: It refers to image plane itself. two categories

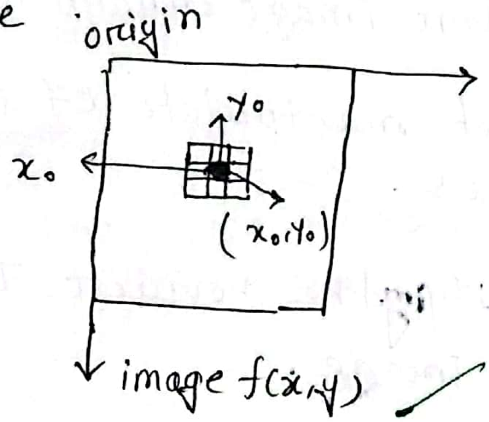
→ Intensity transformation: The modification of intensity values in pixel.

→ Spatial filtering:

The spatial domain on the expression →

$$g(x,y) = T[f(x,y)]$$

where, $f(x,y)$ is an input image $g(x,y)$ is the output image, and T is an operator on f defined over a neighborhood of point (x,y). The operator can be applied to the point of pixels of a single image



here, (x_0,y_0) is an arbitary point in image and the small region is the neighborhood of (x_0,y_0).

The process consist of moving the center of the neighborhood from pixel to pixel.

And applying the operator T to the pixels in the neighborhood to yield and output value at the location. The process starts at the top left of the input image and proceed pixel by pixel in a horizontal scan or vertical scan.

The smallest possible neighbourhood is of size 1x1. Here g depends only on the value of f at a single point (x,y) and T becomes an intensity/gray level mapping transformation function of the form

$$s = T(r)$$

→ The result of applying the transformation to every pixel in f to generate the corresponding pixels in g would be to proud an image of higher contrast than the original, by darkening the intensity levels below k and brightening the levels above k. This technique is called contrast stretching.

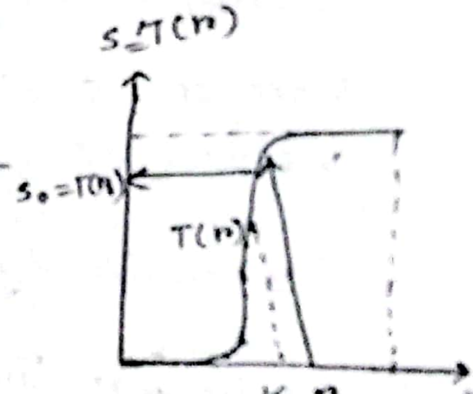


Fig: Contrast stretching

→ Here, T(r) produces a two-level (binary image). A mapping of this form is called a thresholding function.

And applying the operator (T) to the pixels in the neighborhood to yield and output value at the Location. The process starts at the top left of the input image and proceed pixel by pixel in a horizontal scan or vertical scan.

The smallest possible neighbourhood is of size 1X1. Here g depends only on the value of f at a single point (x,y) and T becomes an intensity/gray level/mapping transformation function of the form

$$s = T(r)$$

→ The result of applying the transformation to every pixel in f to generate the corresponding pixels in g would be to proud an image of higher contrast than the original, by darkening the intensity levels below k and brightening the levels above k. This technique is called contrast stretching.

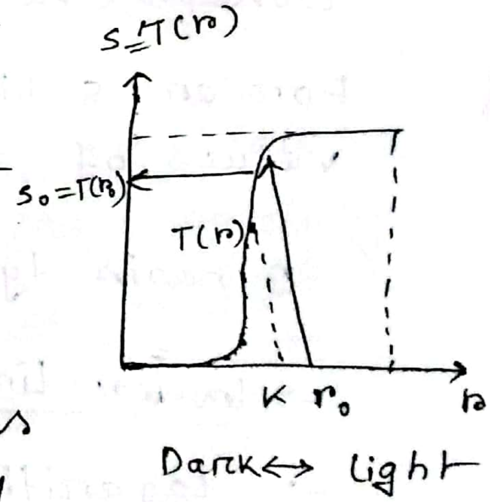
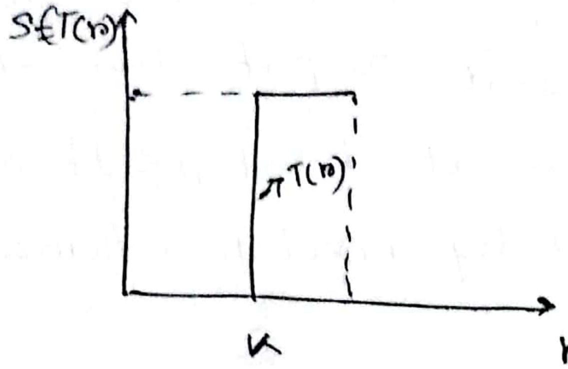


Fig: Contrast stretching

→ Here, T(r) produces a two-level (binary image). A mapping of this form is called a threshbolding function.



Dark ↔ light

Approach whose results depend only on the intensity at a point sometimes are called point processing technique, as opposed to the neighborhood processing technique.

The mapping for r to s are implemented via table lookups (values of intensity transformation function)

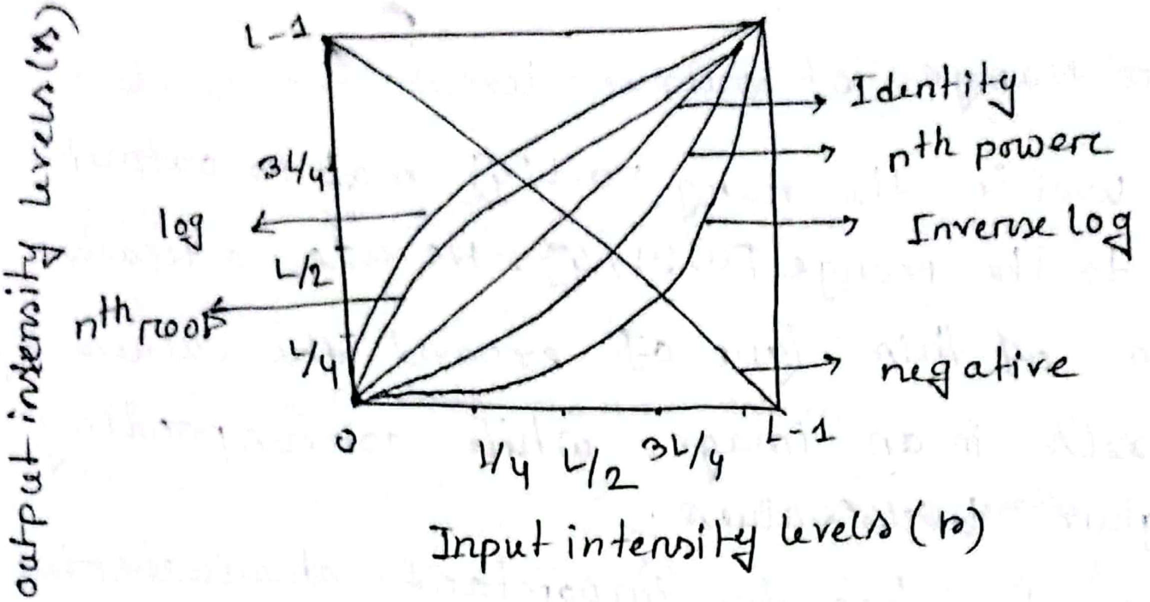
For an 8-bit image, a lookup table cotaining the values of T will have 256 entities.

3 basis types of functions used in image processing

→ ~~linea~~ Linear (negative & identity)

→ Logarithmic (Log & inverse log)

→ Power law (nth powers & nth root transformation)



Linear :

- Identity : The identity function is the trivial case in which the input and output intensities are identical.

- Negatives: The negative of an image with intensity levels with the function is—

$s = L-1-r$

It is used in enhancing white or gray detail embedded in dark regions of an image especially when the black areas are dominant in size.

Log transformation :

The general form of log transformation is -

$$\boxed{s = c\log(1+r)}$$

where, c is a constant and it is assumed that $r \geq 0$, the shape of the log curve shows that this transformation maps a narrow range of low intensity values in the input into widerr range of output levels.

Input level in the rang $[0, L/4]$ map to output levels to the range $[0, 3L/4]$. We use a transformation of this type of expand the values of dark pixels in an image, while corresponding the higher level values.

The Log function has the important characteristic that is compresses the dynamic rang of pixel values.

Power Law transformation:

$s = L - 1 - r$
$s = c\log(1+r)$
$s = cr^{\gamma}$

$$\boxed{s = cr^{\gamma}}$$

$s = c$

Power law curves with fractional values of γ map a narrow rang of dark input values into a wide range of output values with the opposite being true for higher values of input levels.

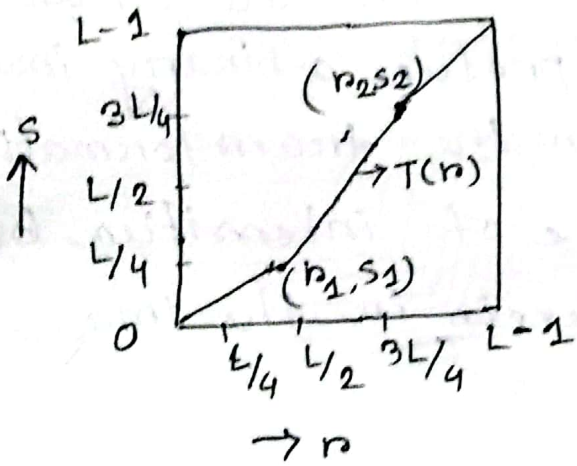
It is also called gamma correction or gamma encoding

Ex: CRT (cathode Ray tube)

cathode Ray tube

Piecewise Linear transformation functions:

(Contrast stretching): For low contrast image, Low contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even the wrong setting of a lens aperture during image acquisition. (Contrast stretching expands the range of the recording medium or display device.



→ make dark position in more low darker and bright portion is more brighter.

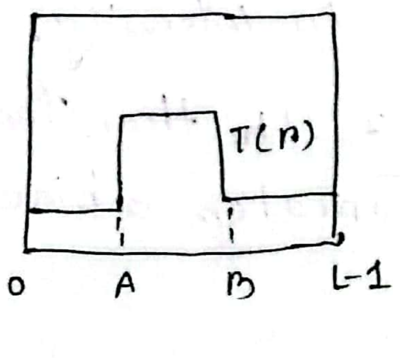
→ If $r_1 = s_1$ & $r_2 = s_2$ then the transformation is a linear function that produces no changes in intensity.

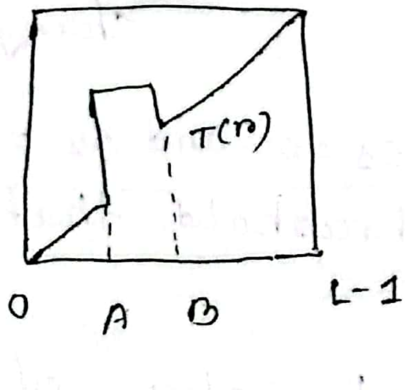
→ If $r_1 = r_2$, $s_1 = 0$ and $s_2 = L-1$, the transformation becomes a threshold that creates a binary image.

→ Intermidiate values of (r_1, s_1) and (r_2, s_2) produce various degrees of spread in the intensity levels of the output image, thus affecting its contrast.

⇒ Intensity - level slicing :

There are application in which it is of interest to highlight a specific range of intensities in an image. Some of these applications include enhancing features in satelite imagery, such as masses of water; and enhancing flows in x-ray images. One approach is to display in one value all the values in the range of interest and in another all other intensities. This transformation produce a binary image. The second approach based on the transformation brighteners the desire range of intensities, but leaves all other intensity levels in the image unchanged.



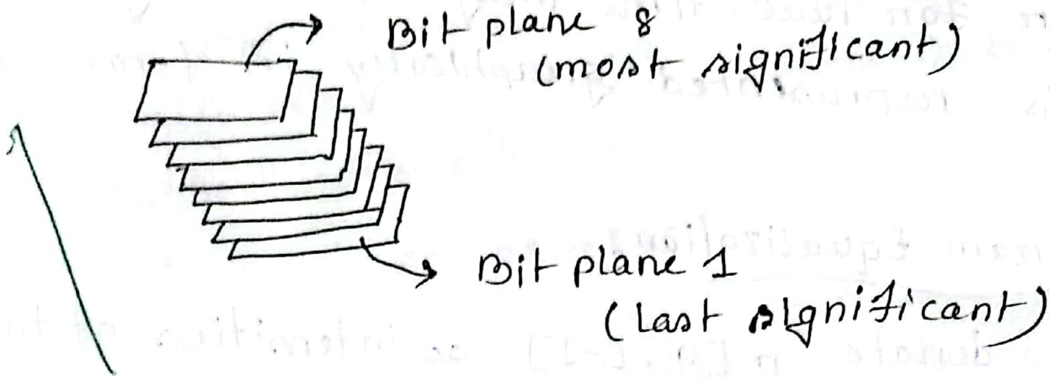


Bit plane slicing :

It is a technique where an image is decomposed into its individual bit planes. Each bit planes represents a single bit of information from each pixel in the original image.

→ 8 bit image → $2^8 = 256$
So intensity level is (0 - 255)

- Convert the image to binary
- Extract bit planes.
- Analyze or manupulate.
- Image compression Application.
- Feature extraction • Watermarking
- Image restoration.



Histogram Processing :→

Histogram . processing → Image Enhancement
→ " Segmentation.
→ " compression

Let r_k, for $k = 0, 1, 2, \ldots L-1$ denote the intensities of an L-level digital image, $f(x,y)$. The unnormalized histogram of f is defined as →

$$h(r_k) = n_k \quad \text{for} \quad k = 0, 1, 2, \ldots L-1.$$

where, n_k is the number of pixels in f with intensity

r_K and the subdivisions of the intensity scale are called histogram ~~thas~~ bins. similary, the normalized histogram of f is defined as

$$P(r_K) = \frac{h(r_K)}{MN} = \frac{n_k}{MN}$$

image rows image colums.

Mostly we work the normalized histograms. The sum of $P(r_K)$ for all values of k is alway 1. Histo- popular for real time image processing. gram is represented graphicaly in form of plot.

Histogram Equalization:

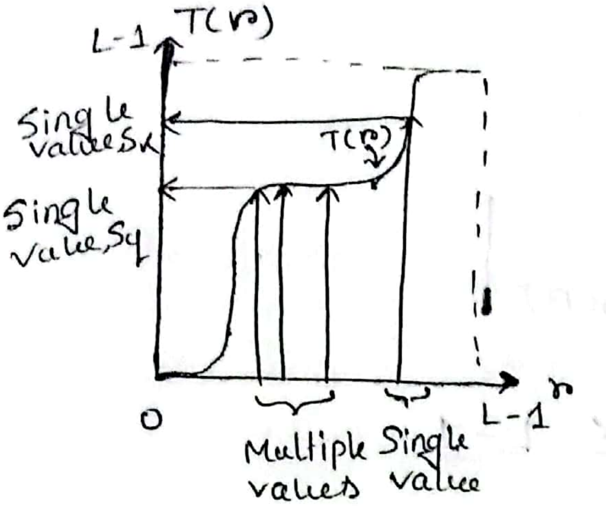
Let use denote r $[0, L-1]$ as intensities of the image to be processed $r=0$ corresponding to black and $r = L-1$ representing white.

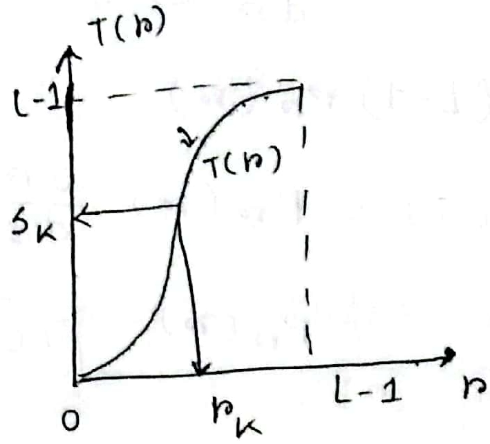
Let the intensity transformation is defined by $s = T(r)$

where, $0 \le r \le L-1$

- $T(r)$ is monotonically increasing function in the interval $0 \le r \le L-1$.
- $0 \le T(r) \le L-1$ and $0 \le r_i \le r \le L-1$.

suppose we use the inverse operation as $r = T^{-1}(s)$ then, the condition should be strictly monotonically increasing.





Strictly monotonically increasing Mapping is one to one in both the directions

- Satisfies the condition $T(r)$ is monotonically increasing function in the interval $0 \leq r \leq L-1$ and $0 \leq T(r) \leq L-1$ and $0 \leq r \leq L-1$

→ Let us consider intensity levels in the image as random variables in the interval 0 to $L-1$.

→ Let us defined the probability Density Function (PDF) as $P_r(r)$ and $P_s(S)$ for r and s respectively

$$P_s(s) = P_r(r) \left| \frac{dr}{ds} \right|$$

where $T(r)$ is continuous and $P_r(r)$ and $T(r)$ is known

The transformation function →

$$s = T(r) = L-1 \int_0^r P_r(w)\, dw$$

grayscale
0 → white
255 → Black

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$

$$= (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$

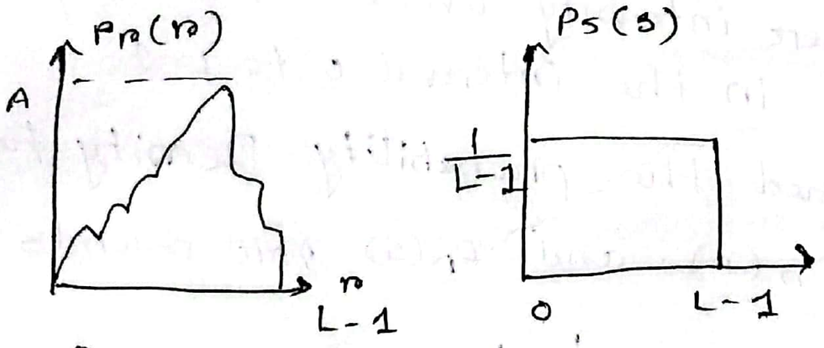
$$= (L-1)\,p_r(r)$$

$$p_s(s) = p_r(r)\,\frac{dr}{ds}$$

$$= p_r(r)\left|\frac{1}{(L-1)\,p_r(r)}\right|$$

$$= \frac{1}{L-1} \qquad 0 \le s \le L-1$$

→ which is uniform probability density function. this means, performing intensity transformation yields a random variable a characterized by uniform PDF.



.0

⇒ For the discrete values of the histogram, we deal with the summation instead of integration

$$p(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, \ldots L-1$$

The discrete form of transformation is given by

$$P(r_K) = \frac{n_K}{MN} \qquad K = 0, 1, 2, \ldots L-1$$

Then do

$$S_K = T(r_K) = (L-1) \sum_{j=0}^{k} P_r(r_j)$$

$$= \frac{(L-1)}{MN} \sum_{j=0}^{k} n_j \qquad k = 0, 1, \ldots L-1$$

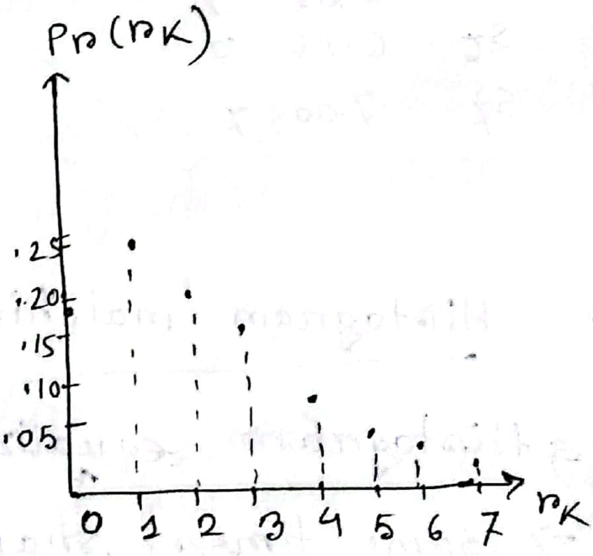
→ The input pixel r_K is mapped to output pixel S_K

→ The transformation (mapping) $T(r_K)$ is called as histogram equalization or

Ex →

Let us consider a 3 bit image ($L=8$) of 64×64 ($MN = 4096$) has the intensity distribution shown below: →

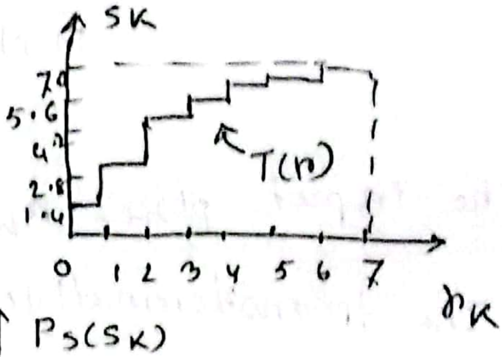
r_K	n_K	$P_r(r_K) = n_K/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



From the equation of histogram equalization, we have,

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

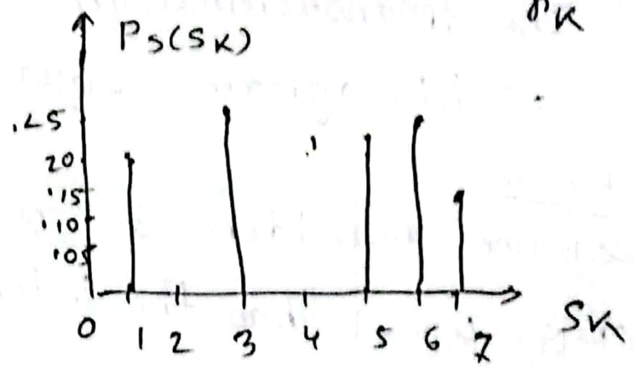


Similarly,

$s_2 = 4.55$ $s_5 = 6.65$

$s_3 = 5.67$ $s_6 = 6.86$

$s_4 = 6.23$ $s_7 = 7.00$



s_0	1.33	1
s_1	3.08	3
s_2	4.55	5
s_3	5.67	6
s_4	6.23	6
s_5	6.65	7
s_6	6.86	7
s_7	7.00	7

Histogram matching (specification) :→

⇒ Histogram equalization is an automatic enhancement

→ some times shape of the histogram can be specified based on the requirement.

→ The method used to generate a processed image that has a specified histogram is called histogram matching or histogram specification

$$p(r_k) = \frac{n_k}{MN} \quad k = 0, 1, 2, \ldots L-1$$

The discrete form of transformation is given by

$$s_K = T(r_K) = (L-1) \sum_{j=0}^{K} p_r(r_j)$$

$$= \frac{L-1}{MN} \sum_{j=0}^{k} n_j \quad K = 0, 1, 2, \ldots L-1$$

Let $P_z(z)$ is the specified PDF, which is going to be the PDF of the output image. So we have,

$$G(z_q) = (L-1) \sum_{j=0}^{q} P_z(z_i) = s_K$$

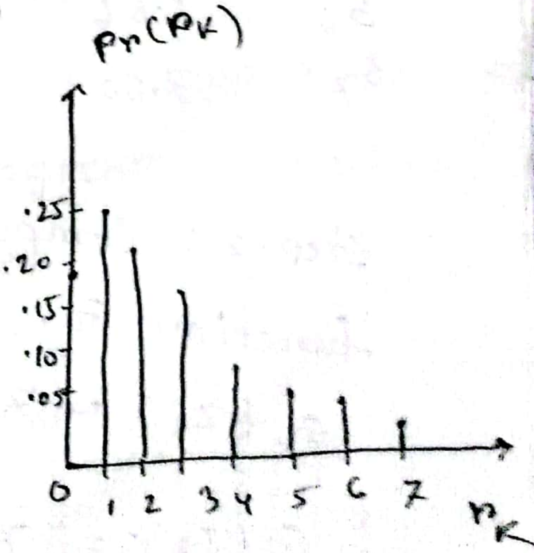
Desired value, $z_q = G^{-1}(s_K)$

This will give value of z for each value of s, by performing mapping of s to z.

Let us understand it by an example –

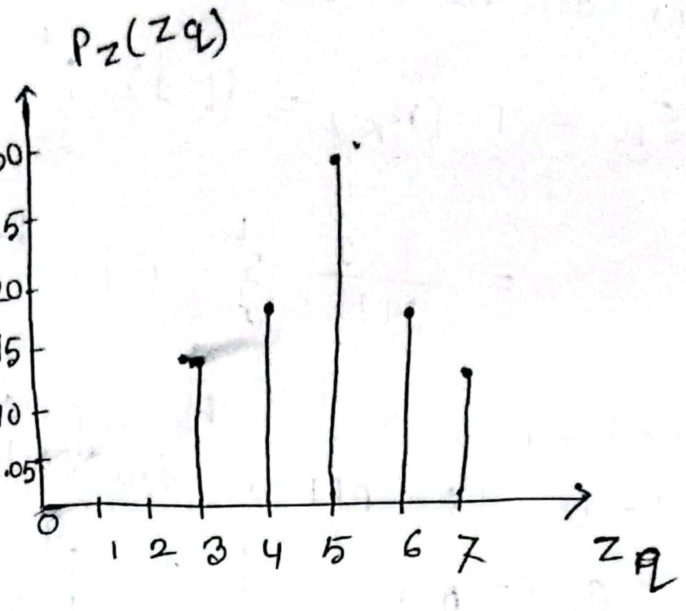
Let us consider a 3 bit image ($L=8$) of 64×64 ($MN = 4096$), has the intensity distribution shown below: →

r_K	n_K	$P_r(r_K) = n_K/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02



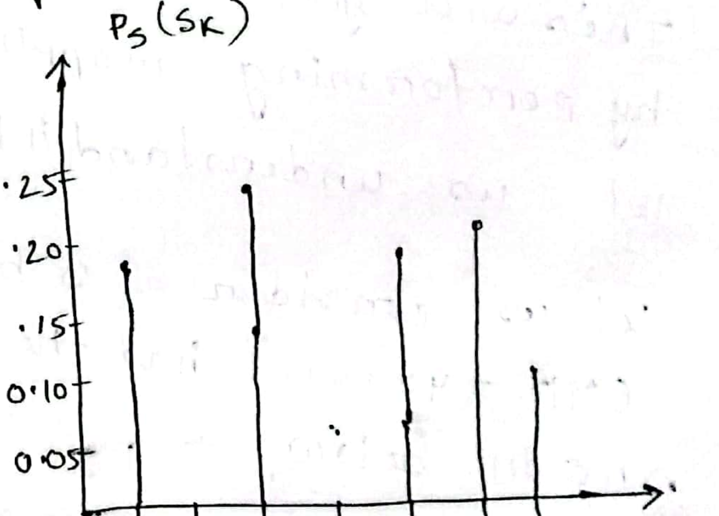
Specified histogram is given as follows :

z_q	$P_z(z_q)$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15



step-1 : Scaled histogram - equalized values

s_0	1.33	1
s_1	3.08	3
s_2	4.55	5
s_3	5.67	6
s_4	6.23	6
s_5	6.65	7
s_6	6.86	7
s_7	7.00	7



step-2 : Compute all the values of transformation function G,

$$G(z_0) = 7 \sum_{j=0}^{0} P_z(z_j) = 0.00$$

$$G(z_1) = 7 \sum_{j=0}^{1} P_z(z_j) = 7[P(z_0) + P(z_1)]$$

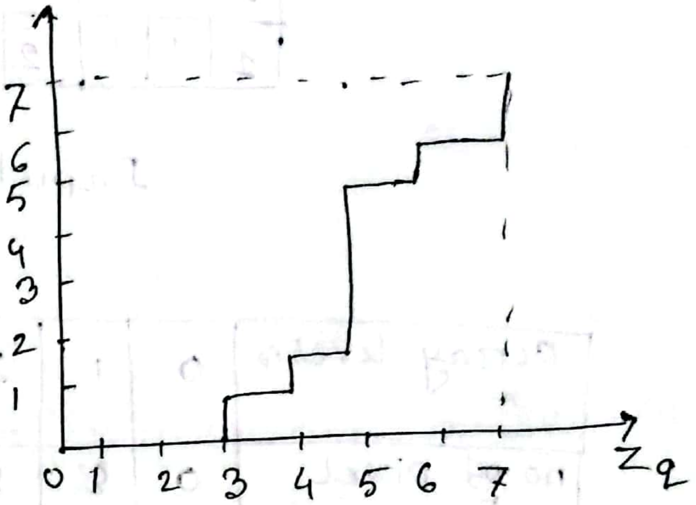
$$= 0.00$$

$G(z_2) = 0.00$ $G(z_3) = 1.05$ $G(z_4) = 2.45$

$G(z_5) = 4.55$ $G(z_6) = 5.95$ $G(z_7) \doteq 7.00$

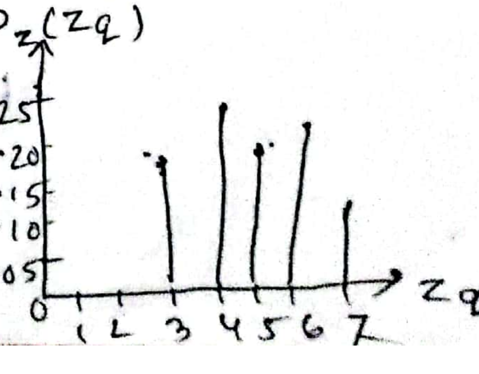
These fractional values are converted to integer values as shown

$G(z_0)$	0.00	0
		0
$G(z_1)$	0.00	0
$G(z_2)$	0.00	0
$G(z_3)$	1.05	1
$G(z_4)$	2.45	2
$G(z_5)$	4.55	5
$G(z_6)$	5.95	6
$G(z_7)$	7.00	7



→ The condition of strictly monotonic is violated
To handle this situation following procedure is used
For example $s_0 = 1$, and $G(z_3) \doteq 1$, which is
perfect match for this case, here $s_0 \to z_3$,
i.e. every pixel whose value is 1 in the
histogram equalized image is mapped to
pixel valued 3 in the histogram specified
image. Continuing this we get.

s_k	z_q
1	3
3	4
5	5
6	6
7	7



Q2: Perform histogram equalization for the following image

$$f(x,y) = \begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 1 & 1 & 1 \\ \hline 2 & 5 & 3 & 5 & 2 \\ \hline 2 & 5 & 5 & 5 & 2 \\ \hline 2 & 5 & 3 & 5 & 2 \\ \hline 1 & 1 & 1 & 2 & 1 \\ \hline \end{array}$$

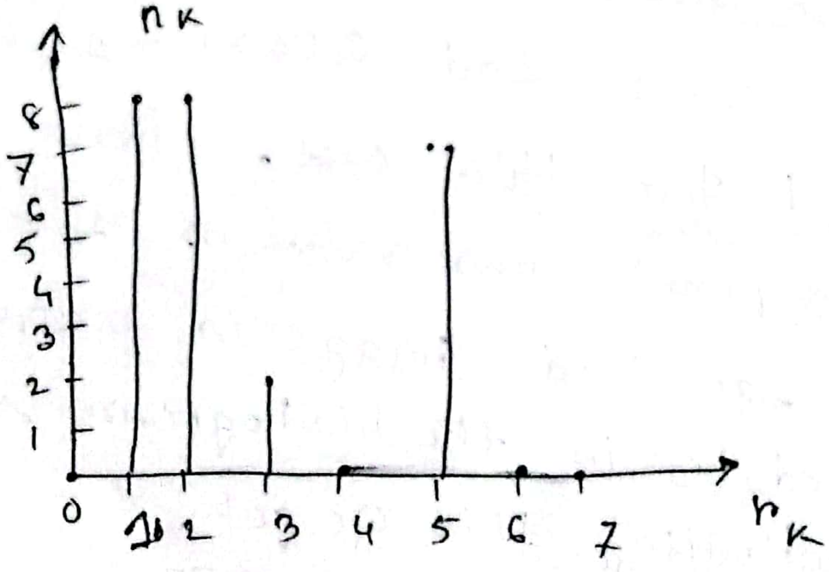
max value = 5

Input Image

Gray levels	0	1	2	3	4	5	6	7
no of pixel n_k	0	8	8	2	0	7	0	0

$L-1 = 8-1 = 7$

highest gray level value is 5. so, $2^3 \rightarrow 3$ bits. needed ; Maximum gray level value = 7



Histogram of input image.

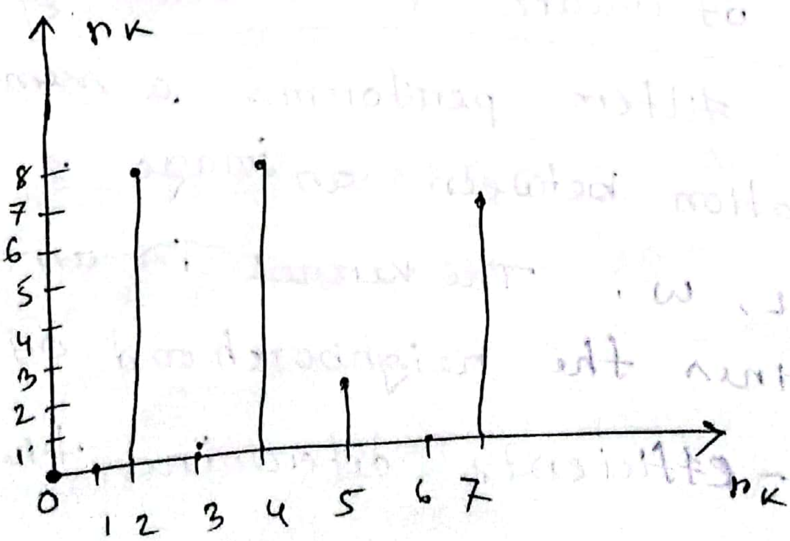
Gray level	no of pixel	PDF= n_k/MN	CDF s_k	$s_k \times 7$	Histogram equalization level
0	0	0	0	0	0
1	8	0.32	0.32	2.24	2
2	8	0.32	0.64	4.48	4
3	2	0.08	0.72	5.04	5
4	0	0	0.72	5.04	5
5	7	0.28	1	7	7
6	0	0	1	7	7
7	0	0	1	7	7

$n = 25$

For discrete form of the tranformation equ. is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

Gray levels	0	2	4	5	7
No. of pixel	0	8	8	2	7



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

→

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

output image

Fundamental of Linear spatial filtering :→

~~A linear~~ spatial filtering modifies ~~an~~ image by replacing the value of each pixel by a function of the value of the pixel and its neighbors, If the operation performed on the image pixel is linear then the filter is called a linear spatial filter.

The mechanics of Linear spatial filtering :

A linear spatial filter performs a sum of product operation between an image f and a filter kernel, w. The kernel is an array whose size defines the neighborhood of operation and whose co-efficients determine the nature of the filter.

Spatial kernel are also called mask template window:

For 3 X 3 kernel, at any point (x, y) in the image, the response g(x, y) —

w(-1,-1)	w(-1,0)	w(-1, 1)
w(0,1)	w(0, 0)	w(0, 1)
w(1,-1)	w(1, 0)	w(1, 1)

kernel co-efficients

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

pixel values under kernel when it is centered on (x, y)

$$g(x,y) = w(-1,-1)\, f(x-1, y-1) + w(-1,0)\, f(x-1, y) + \cdots\ w(1,1)\, f(x+1, y+1)$$

m x n ⟹ m = 2a+1 and n = 2b+1

here, a, b are non-integer value. if a = 1, b = 1

so, m = 3, n = 3. So minimum mask is 3 x 3

→ Linear filtering of image of size M X M and mask size m X m

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

where, $a = \frac{m-1}{2}$ and $b = \frac{n-1}{2}$

correlation/ convolution

→ correlation/ convolution is the process of moving a filter mask over the image and computing the some of products at each location.

→ We use correlation to check similarity between two images.

- Difference between convolution and co-rrelation is that the convolution process rotates the matrix by 180 degrees.

EX: origin (f)

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

(a)

padded f

0 0 0 0 0 0 0
0 0 0 0 0 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 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

(b)

mask (w)

1 2 3
4 5 6
7 8 9

Initial position for w

1 2 3 0 0 0 0
4 5 6 0 0 0 0
7 8 9 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

(c)

Correlation result

0 0 0 0 0
0 9 8 7 0
0 6 5 4 0
0 3 2 1 0
0 0 0 0 0

(d)

Full correlation result

0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 9 8 7 0 0
0 0 6 5 4 0 0
0 0 3 2 1 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

(e)

Rotated w

9 8 7 | 0 0 0 0
6 5 4 | 0 0 0 0
3 2 1 | 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

(f)

Convolution

0 0 0 0 0
0 1 2 3 0
0 4 5 6 0
0 7 8 9 0
0 0 0 0 0

(g)

full convolution result

0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 1 2 3 0 0
0 0 4 5 6 0 0
0 0 7 8 9 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

(h)

Smoothing (lowpass) spatial filters:

→ Smoothing spatial filters are generally used for - blurring and noise reduction.

→ Blurring is generally used in preprocessing, such as removal of small details from an image prior to object extraction, and bridging of small gaps in lines or curves.

→ Noise reduction can be acomplished by blurring with a linear filter and also by non-linear filtering.

⇒ Smoothing Linear Filters:

→ The output of a smoothing by linear filter is simply the average of the pixels contained in the neighborhood of the filter mask

→ These filter sometimes are called averaging filters, low pass filters

→ The process of replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask result in an image with reduce 'sharp' transition in gray levels.

→ The process also includes the smoothing of false contours that result from sufficient number of gray levels.

→ Therefore, averaging filters mostly reduce "irrelevant" detail in an image.

→ 'irrelevant' have refers to pixel regions are small with respect to the size of the filter mask

i. Box filter – all co-efficient are equal

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}_{3\times3} \rightarrow \text{mask}$$

ii. Weighted average – give more (less) weight to pixels near the output location

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

non-linear filters:

Their response is based on ordinary the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result.

i. Median filter – Find the median of all the pixel values.

ii. Min filter – Find the minimum of all the pixel values

iii. Max filter – Find the maximum of all the pixel values.

Box filter:

Let us consider the below image matrix for understanding and to do the operation of noise removal with the box filter.

(Here, we assume the pixels which are identified as noise alone for the operation. Else it would actually be moving row * column impact each cell) :

$$\begin{bmatrix} 50 & 50 & 50 & 100 & 100 \\ 50 & 58 & 50 & 100 & 100 \\ 50 & 50 & 50 & 100 & 100 \\ 50 & 61 & 50 & 110 & 100 \\ 50 & 50 & 50 & 100 & 100 \end{bmatrix}$$

The input image matrix – with noise plotted

→ The filter to be used for the suppression is presented matrix below:

$$1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The Box Filter of size 3×3.

- The region of consideration from the 5×5 matrix is presented below. In that 3×3, 58 is the noise pixel.. The below procedure shall enable the readers to understand how to eliminate the noise with the box filter operations:

$$\begin{bmatrix} 50 & 50 & 50 & 100 & 100 \\ 50 & 58 & 50 & 100 & 100 \\ 50 & 50 & 50 & 100 & 100 \\ 50 & 61 & 50 & 110 & 100 \\ 50 & 50 & 50 & 100 & 100 \end{bmatrix} \text{ conv } 1/9 * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$f(1,1) = 1/9\,(50\times1 + 50\times1 + 50\times1 + 50\times1 + 58\times1 + 50\times1 + 50\times1 + 50\times1 + 50\times1)$$

$$= 458/9 = 50.8$$ approximated to 51

- Hence 51 has to replace 58. The noise is suppressed by a large difference. This is achieved all through the neighborhood calculation as shown above.

- As one more instance, we could suppress the noise for f(3,1) the pixel value 61 has to be suppressed

$$\begin{bmatrix} 50 & 50 & 50 & 100 & 100 \\ 50 & 58 & 50 & 100 & 100 \\ 50 & 50 & 50 & 100 & 100 \\ 50 & 61 & 50 & 100 & 100 \\ 50 & 50 & 50 & 100 & 100 \end{bmatrix} \text{ conv } \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$f(3,1) = 461/9 = 51.2251$

Hence the new update values for f(1,1) f(3,1) respectively is 51 and the image matrix gets restructured as:-

$$\begin{bmatrix} 50 & 50 & 50 & 100 & 100 \\ 50 & \boxed{51} & 50 & 100 & 100 \\ 50 & 50 & 50 & 100 & 100 \\ 50 & \boxed{51} & 50 & 110 & 100 \\ 50 & 50 & 50 & 100 & 100 \end{bmatrix}$$

In the similar way, the rest of the operation can be carried out.

Sharpening Spatial filters: →

→ The objective of sharpening is the highlight fine detail in an image or to enhance detail that has been blurred.

→ Image sharpening include applications ranging from electronic printing and medical imaging

→ Image to industrial inspection and autonomous guidance in military systems.

→ Image blurring could be acomplished in the spatial domain by pixel averaging in a neighbor hood. Hence, it is logical to conclude that sharpening could be acomplished by spatial differentiation.

→ Fundamentally, the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

→ This image differentiation enhances edges and other discontinuities and demphasizes areas with slowly varying gray level values

→ A basic definition of the first order derivative of a one dimentional function f(x) is the difference.

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

→ Similarly, we define a second order derivatives as difference $\frac{\delta^2 f}{\delta x^2} = f(x+1) + f(x-1) - 2f(x)$

→ We shall consider an image function of two variables $f(x, y)$

A second order derivative is much more aggressive that a first order derivative in enhancing sharp changes.

Frequency Domain

In spatial domain, directly pixel value is manipulated but in frequency domain firstly image convert to spatial domain then apply F.T and get frequency domain. If we want to restore the image, then apply inverse F.T.

1. Dimentional Discrete F.T:

$x(n) \xleftrightarrow{DFT} X(K)$

Spatial — Frequency

N = no of sample

$$X(K) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi/N\, kn} \; ; \; 0 \le K \le N-1$$

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K)\, e^{j2\pi/N\, kn} \; ; \; 0 \le n \le N-1$$

Fourier Spectrum:

$$|F(U)| = [R^2(u) + I^2(U)]^{1/2}$$

phase Angle:

$$\phi(U) = \tan^{-1}\left[\frac{I(V)}{R(U)}\right]$$

power spectrum,

$$F(U) = |F(U)|^2 = R^2(U) + I^2(U)$$

2. Fourier transform of one continuous variable

The fourier transform of a continuous function $f(t)$ of a continuous variable, t denoted $\mathcal{F}\{f(t)\}$ is defined by the eq.

$$\mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi \mu t}\, dt$$

where μ is continuous variable

$$\mathcal{F}\{f(t)\} = F(\mu)$$

$$\therefore F(\mu) = \int_{-\infty}^{\infty} f(t)\, e^{-j2\pi \mu t}\, dt$$

The inverse fourier transform,

$$f(t) = \int_{-\infty}^{\infty} F(\mu)\, e^{j2\pi \mu t}\, d\mu$$

2D Discrete F.T :

$$f(x,y) \rightleftarrows F(u,v)$$

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-2j\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

Basic steps for filtering in the frequency domain →

1. Given an input image $f(x,y)$ of size $M \times N$, obtain padding sizes P and Q, that is $P = 2M$, and $Q = 2N$.

2. Form a padded image $f_p(x,y)$ of size $P \times Q$ using zero, mirror or replicate padding

3. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center the F.T on the $P \times Q$ frequency rectangle.

4. compute the DFT, $F(u,v)$ of the image

5. construct a real, symmetric filter transfer function, $H(u,v)$ of size $P \times Q$ with center at $(P/2, Q/2)$

6. Form the product $G(u,v) = H(u,v) , F(u,v)$ using element wise multiplication; that is $G(i,k) = H(i,k) \; F(i,k)$ for $i = 0, 1, 2, \ldots M-1$ and $k = 0, 1, 2, \ldots N-1$

7. obtain the filtered image by computing the IDFT of $G(u,v)$ $g_p(x,y) = ($real

$[\mathcal{F}^{-1}\{G(u,v)\}]) \, (-1)^{x+1}$.

8. Obtain the final filtered result, $g(x,y)$ of the same size as the input image.

Spatial Domain

$f(x,y) \rightarrow \boxed{h(x,y)} \rightarrow g(x,y)$

$g(x,y) = h(x,y) * f(x,y)$

freq Domain

$F(U,V) \rightarrow \boxed{H(U,V)} \rightarrow G(u,v)$

$G(U,V) = F(U,V) \times H(U,V)$

$g(x,y) = F^{-1}(F(U,V) \cdot H(U,V))$

Image smoothing in freq. Domain

- high pass (cut of low freq.)
- Low pass (cut of high freq)
  - Ideal L.P.F
  - Butterworth L.P.F
  - Gaussian LPF

Ideal Low pass Filter:→

The ideal low pass filter, which is commonly regarded as ILPF is the most used filter for image smoothing in the frequency domain. As expected, the ILPF removes the high frequency content (noise) from the image and retains the low frequency components

Once could mathematically represented the low pass filter through the function :

$$H(U,V) = \begin{cases} 1 & D(U,V) \le D_0 \\ 0 & D(U,V) > D_0 \end{cases}$$

• D_0 – positive constant. The filter retains all the frequency components within the radius D_0

• All the frequency components outside the radius of the circle shall be eliminated. Most of the cases it would ~~all~~ be all the high frequency components getting eliminated. Within the circle, the retention happens without attenuation

• D_0 is the one referred as the cut off frequency

• Also $D(u,v)$ is the Euclidean Distance from any point (u,v) to the origin of the frequency plane and the center of the

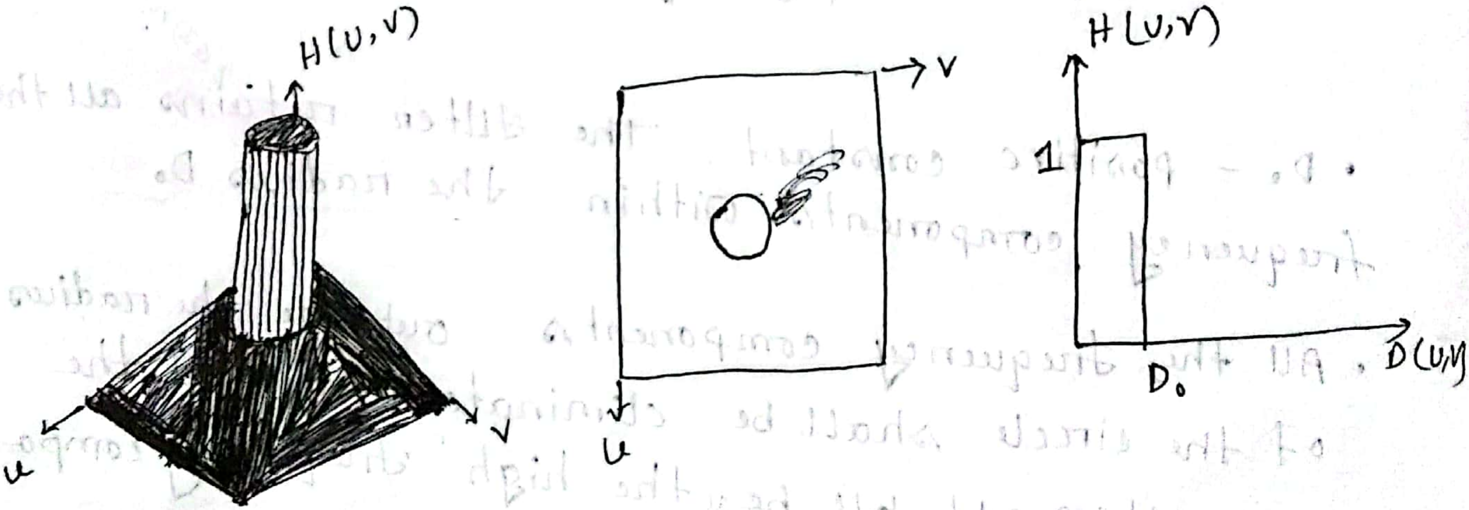
P×Q frequency rectangle, that is,

$$D(u,v) = \left[(u-P/2)^2 + (v-Q/2)^2 \right]^{1/2}$$

where, P and Q are the padded size.

⇒ One should note that, if the value of D_0 is very minimal, there is a risk of losing the core information and it would be over blurred or smoothened. Hence, it is important to choosen the apt D_0 value!

যথাযথ



Butterworth low pass filter:

→ Butterworth low pass filter is something special. It is used to get as flat as frequency response as possible.

→ It is normally said to be used for the image smoothing when it comes to the frequency domain.

→ Butterworth low pass filter, often recognized as BLPF is very helpful in removal of the high frequency noise from the input image while preserving the low frequency contents.

→ Butterworth filter's frequency response is to be noted. It has no sharp frequency response transition. When compared with the ideal low pass filter BLPF is said to have a very smooth transition.

→ The transition between the stop-band and pass band is totally determined by the order of the filter.

→ Here, we represent could be soft and gentle or it can as well be abrupt and sudden. So based on the value of n, one could determine the transition.

The transfer function of a BLPF of order n, with cutoff frequency at a Distance D_0 from the center of the frequency rectangle.

$$H(U,V) = \frac{1}{1+[D(U,V)/D_0]^{2n}}$$

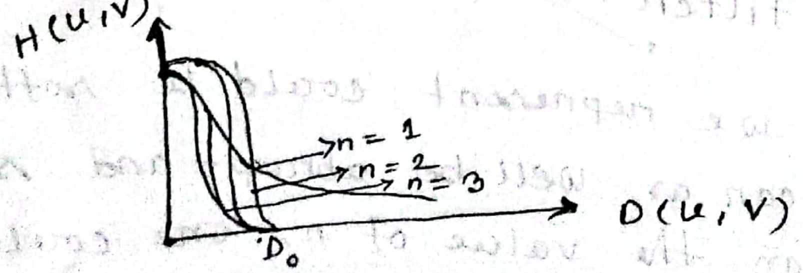
The transition point value between 1 and 0 ($H(U,V)=1$ and $H(U,V)=0$). It work based on the D_0 value. BLPF is ~~som~~ smooth transition happening between 1 and 0. This is the reason BLPF is said to be fight and ringing effects:

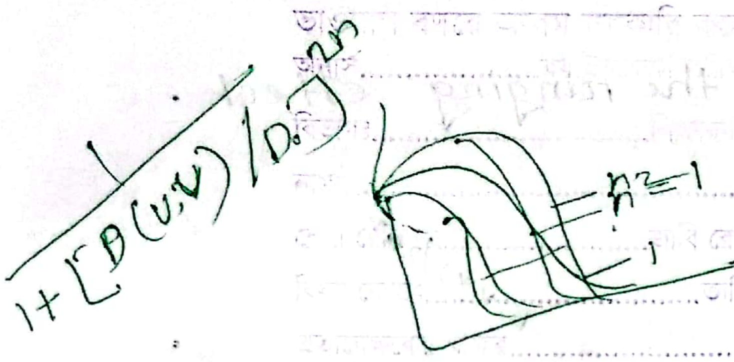
→ n - will define order.

→ If n is increases, Filter becomes sharper but ringing effect.

→ If $n=1$; no ringing effect exist

→ If $n=2$ ringing is present but imperceptible





Gaussian Low pass Filter:

→ Gaussian Filters are very important in many signal processing, image processing and communication applications.

→ These filters are characterised by narrow bandwidths, sharp-cut-offs, and low overshoots.

→ A key feature of Gaussian filter is that the FT of a Gaussian is also a Gaussian.

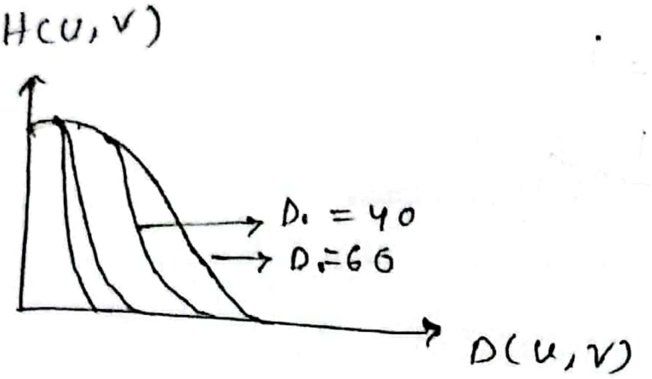
→ So the filter has the same response shape in both spatial and frequency domains.

→ Where D(u,v) is the Distance from the origin in the frequency plane. The parameter a measures the spread of the Gaussian curve.

→ Large the value of the a, larger the cut-off frequency.

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2} \quad \Big| \quad \sigma = D_0$$

It is used to avoid the ringing effect.



$e^{-D^2(u,v)/2\sigma^2}$

Image sharpening using frequency domain

i. Ideal high pass filtering

ii. Butterworth high pass filtering

iii. Gaussian " "

We use for sharpening high pass filter.

$H(u,v)$ of size $p \times q$

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

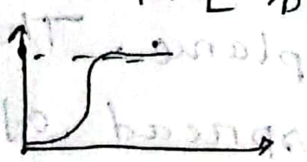
IHPF:

$$H(u,v) = \begin{cases} 0, & \text{if } D(u,v) \le D_0 \\ 1, & \text{if } D(u,v) > D_0 \end{cases}$$

D_0 = cut off freq.

$D(u,v)$ = Distance from the center of the freq. rectangle

BHPF: $H(u,v) = \dfrac{1}{1 + \left[D_0 / D(u,v) \right]^{2n}}$



GHPF: $H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$

Laplacian in the frequency Domain:

$$H(u,v) = -4\pi^2(u^2+v^2)$$

Filters in frequency domain :

→ It is used to smoothing and sharpening of image by removing very high or low freq. components

* high pass
ii. low pass.

* low pass	high pass
1. remove high freq. component keep low freq. component	1. remove low frequency component keep high freq. component.
2. Smoothing the image	2. Sharpening image
3. Remove noise	3. no background

low pass Filter	High pass filter
1. ideal low pass	1. ideal high pass
2. Butterworth " "	2. Butterworth " "
3. Gaussian " "	3. Gaussian " "

i. Ideal low pass :
Transformation func →

$$H(u,v) = \begin{cases} 1, & D(u,v) \leq D_0 \\ 0, & D(u,v) > D_0 \end{cases}$$

where, D_0 = non-negative integer

$$D(u,v) = \left[(u-M/2)^2 + (v-N/2)^2\right]^{1/2}$$

i. IHF :

$$H_H = 1 - H_L$$

$$= \begin{cases} 1, & D(u,v) > D_0 \\ 0, & D(u,v) \leq D_0 \end{cases}$$

BLPF :

Transformation func →

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

BHPF :

$$TF = 1 - H_L$$

$$= 1 - \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

GLPF :

TF,

$$H_L(u,v) = e^{-D^2(u,v)/2D_0^2}$$

GHPF :

TF,

$$H_H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Application of Image Enhancement →

→ Helps to improve image clarity

→ Enhances image detail visibility

→ Increases the lightness of an image

→ Image smoothing and sharpening

→ Noise Removal

Spatial Domain	Frequency Domain
• operates directly on the pixel values of the image.	• Analyzes the frequency components of the image using mathematical transformation.
• Enhancement techniques involve modifying the pixel values using various algorithm	• Enhacement techniques involve manipulating the frequency spectrum of the image.
• common spatial domain techniques include histogram equalization, contrast stretch-ing and spatial filtering	• common frequency domain techniques includ Fourier Transform, frequency filtering and image restoration.
• These techniques are relatively simple and easy to implement.	• These techniques are more complex and requir advanced mathematical operations.
• They are effective for enhancing local details and improving overall image quality	• They are effective for removing noise, sharpen-ing images and perf-forming image resto-ration tasks

Overall, spatial domain techniques focus on directly modifying the pixel values of an image while frequency domain techniques analyze and manipulate the frequency components of the image. Both approaches have their strengths and weaknesses, and the choice of technique depends on the specific image enhancement requirements.

Colour Image Processing

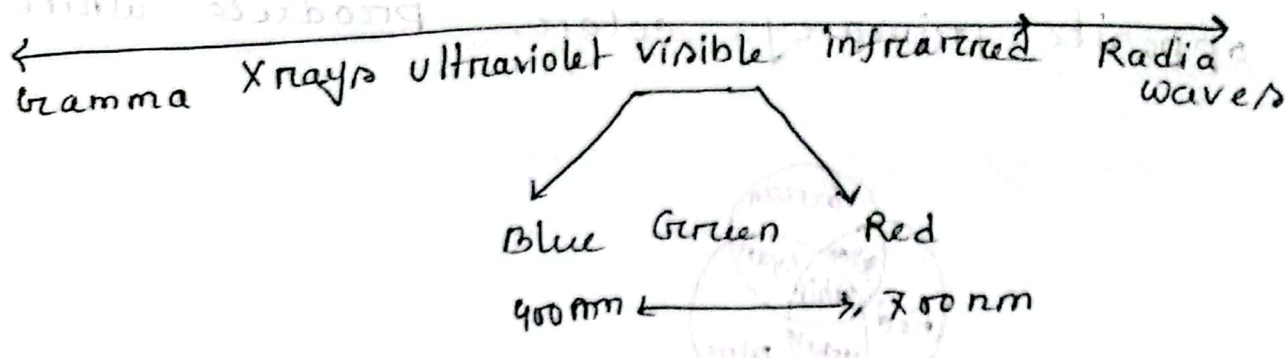
→ Color is a powerful descriptor that simplifies object identification and extraction.

→ Human can discern thousands of color shades and intensities, compared to about only two dozen shades of gray.

Color image processing is divided into two major areas: →

i. Full-color processing: image are acquired with a full colour sensor, such as TV camera or color scanner.

- Psedocolor processing: The problem is one of assigning a color to a particular monochrome intensity or rang of intensities.



Radiance = total amount of energy flow from a light source (watts)

Luminance = amount of energy received by an observer (lumens)

Brightness = Intensity

Human Eye = 65% Red,
33% Green,
2% Blue.

Color Fundamentals:

Color are seen as variable combination of the primary colors of light: red (R), green (G) and blue (B). The primary color can be mixed to produce the secondary

colors: megenta (red + blue)
cyan (green + blue), and
yellow (red + green).

Mixing the three primaries, or a secondary with its opposite primary color, produce white light.

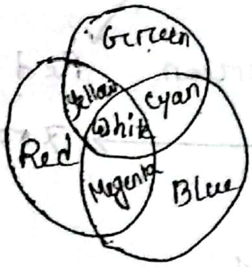
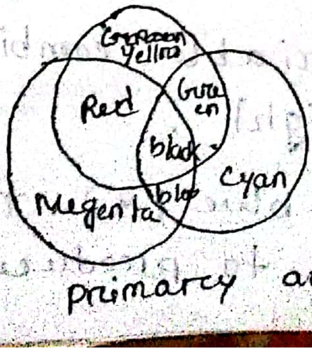


Fig: Primary and secondary colors of light

RGB colors are used for color TV, monitors and vedio cameras.

However, the primary colors are of pigments are cyan (c), megenta (M) and Yellow (Y) and the secondary colors are Red, green and blue.
A proper combination of the three pigment primaries, or a secondary, with its opposite primary, produces black



Primary and secondary colors of pigments

CMY colors are used for color printing.

Color characteristics:

The characteristics used to distance one color from another are:

- Brightness: means the amount of intensity (i.e. color level)
- Hue: represents dominant colors are perceived by an observer.
- Saturation: refers to the amount of white light mixed with a hue.

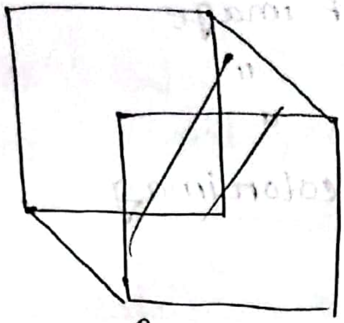
Color Models:

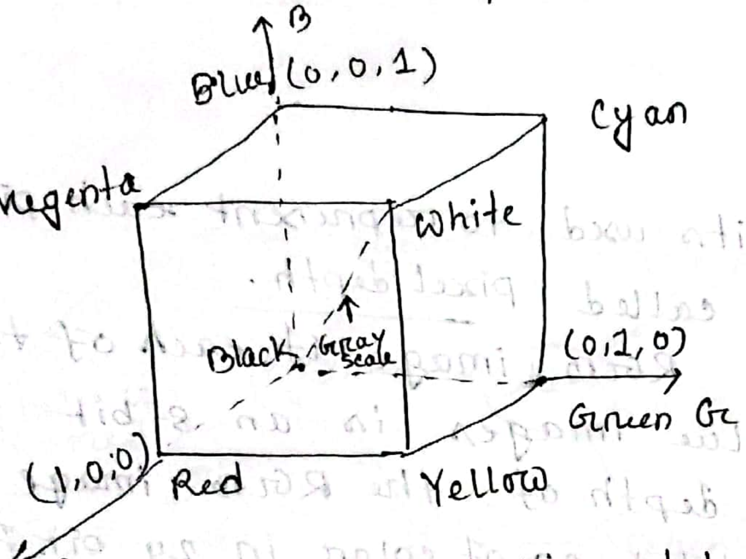
The purpose of a color model is to facilitate the specification of colors in some standard way. A color model is specification of a co-ordinate system and a subsequency within that system where each color is represented by a single point. Color models most commonly used in image processing are:

- RGB model for color monitors and video cameras.
- CMY model and CMYK models (cyan, Megenta, Yellow, black) models for color printing
- HSI (hue, saturation, intensity) model.

⇒ The RGB color model: →

In this model, each color appears in its primary colors red, green and bule. This model is based on a catesian coordinate system. The color subspace is the cube shown in the figure below. The different colors in this model are points on and are defined by vectors extending from the origin.





RGB color model.

All color values R, G, B have been normalized in the range [0, 1].

However, we can represent each of R, G and B from 0 to 255.

Each RGB color image consists of three component images, one for each primary color as shown in figure below. These three image are combine on the screen to produce a color image.

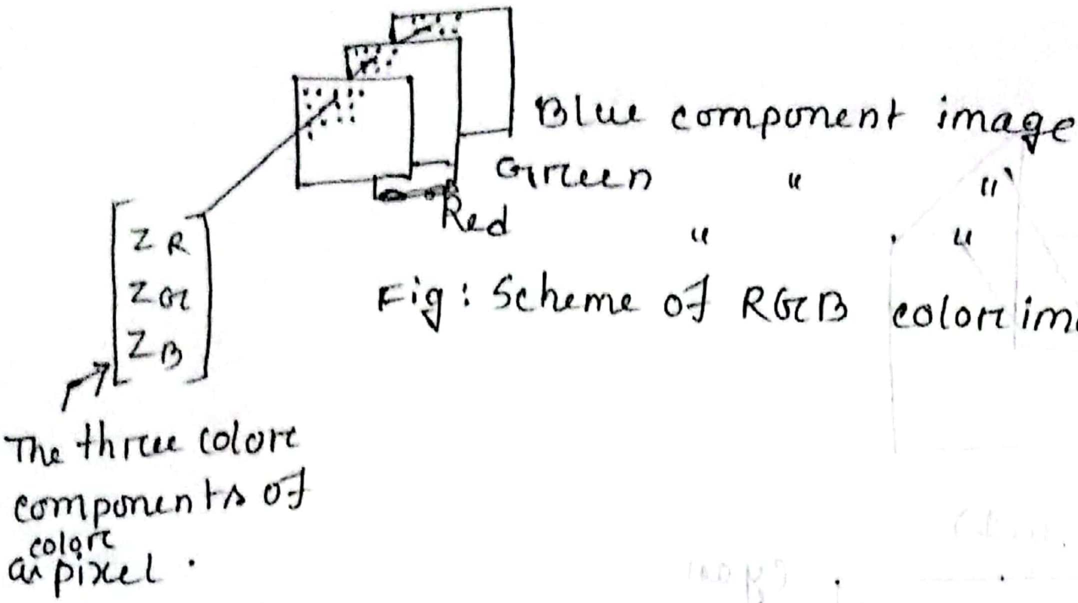


Fig: Scheme of RGB color imag

The total number of bits used to represent each pixel in RGB image is called pixel depth.
For example, in an RGB image if each of the red, green and blue images is an 8-bit image, the pixel depth of the RGB image is 24-bits. The total no of color in 24 bit RGB image is $2^{24} = 16777216$ colors.

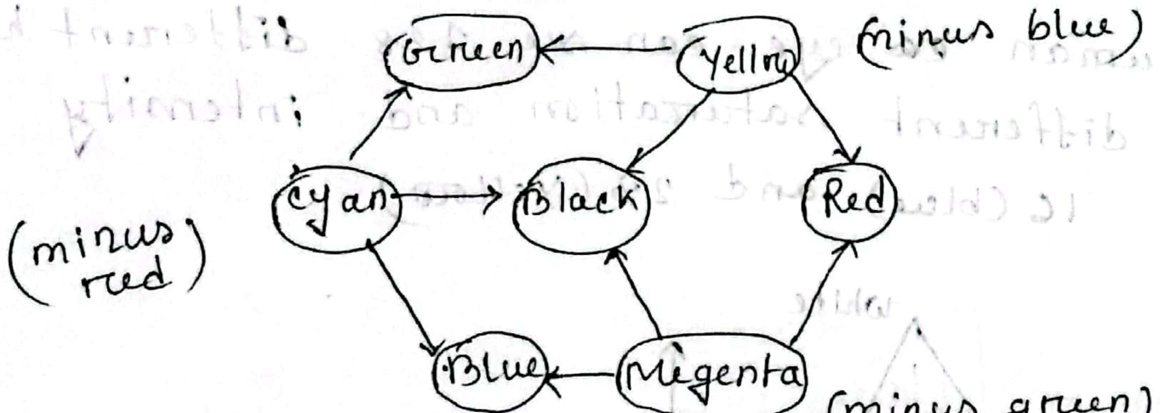
The CMY and CMYK color Model:

cyan, megenta, and yellow are the primary colors of pigments. Most printing devices such as color printer and copiers required CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using the equation —

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

where, all color values have been normalize to the range [0, 1].

In printing, combining equal amounts of cyan, megenta, and yellow produce muddy-looking black. In order to produce true black, a fourth color, black is added, giving a rise to the CMYK color.

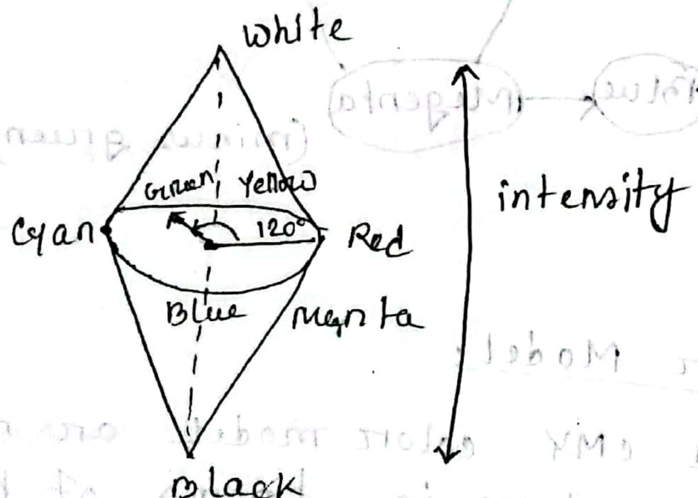


The HSI color Model:

The RGB and CMY color model are not suited for describing colors in terms of human interpretation. When we view a color object we describe it by its hue, saturation and brightness (intensity). Hence the HSI color model has been presented. The HSI model decouples the intensity component from the color-crying information in a color image. As a result, this model is an ideal tool for developing color image processing algorithm.

The hue, saturation and intensity values can be obtained from the RGB color cube. That is, we can convert any RGB point to a corresponding point in the HSI color model by working out the geometrical formulas.

The human ~~ea~~ eye can see 128 different hues, 130 different saturation and intensity been 16 (blue) and 23 (yellow).



HSI to RGB :

→ RGB Sector $(0 \leq H \leq 120)$

$$B = I(1-S)$$

$$R = I\left[1 + \frac{S \cos H}{\cos(60° - H)}\right]$$

$G = 3I - (R+B)$

GB sector $(120° \leq H < 240°)$

$\rightarrow H = H - 120°$

$R = I(1-S)$

$$G = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

$B = 3I - (R+G)$

BR sector $(240° \leq H < 360°)$

$H = H - 240°$

$G = I(1-S)$

$$B = I\left[1 + \frac{S\cos H}{\cos(60° - H)}\right]$$

$R = 3I - (G+B)$

Converting RGB to HSI

The hue H is givenby.

$$H = \begin{cases} \theta, & \text{if } B \leq G \\ 360 - \theta, & \text{if } B > G \end{cases}$$

where,

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}}\right\}$$

The saturation S is given by,

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right]$$

The intensity I is given by

$$I = \frac{1}{3}(R+G+B)$$

All RGB values are normalized to the range [0,1]

Ex: RGB (29, 104, 215)

$R_{nor} = 29/255 = 0.113$

$G_{nor} = 104/255 = 0.407$

$B_{nor} = 215/255 = 0.843.$

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(0.113-0.407)+(0.113-0.843)\right]}{\left[(0.113-0.407)^2+(0.113-0.843)(0.407-0.843)\right]^{\frac{1}{2}}}\right\}$$

$$= 143.6^\circ$$

$\therefore H = 360^\circ - 143.6 = 216.4^\circ$

$$S = 1 - \frac{3}{.113+.407+.843}\left[\min(0.113, 0.407, 0.843)\right]$$

$$= 1 - \frac{3}{1.363} \times 0.113$$

$$= 0.752$$

$I = 0.454$

Basics of Full-color image processing →

Full color image processing approaches fall into two major categories:

- Approach that process each component image individually and then form a composite processed color image from the individually processed components.
- Approach that work with color pixels directly

In full color images, color pixels really are vectors.

For example, in the RGB system, each color pixel can be expressed as

$$c(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

For an image size $M \times N$, there are MN such vectors, $c(x,y)$ for $x = 0, 1, 2, \ldots M-1$
$y = 0, 1, 2, \ldots N-1$

color Transformation →

As with the gray-level transformation, we model color transformations using the expression

$$g(x,y) = T[f(x,y)]$$

where, $f(x,y)$ is a color input image, $g(x,y)$ is the transformed color output image and T is the color transform.

$$s_i = T_i(r_1, r_2, \ldots r_n) \qquad i = 1, 2, \ldots n$$

Image Restoration

Image restoration attempts to restore/reconstruct/recover images that have been degraded.

→ It identifies the degradation process and attemp to reverse it.

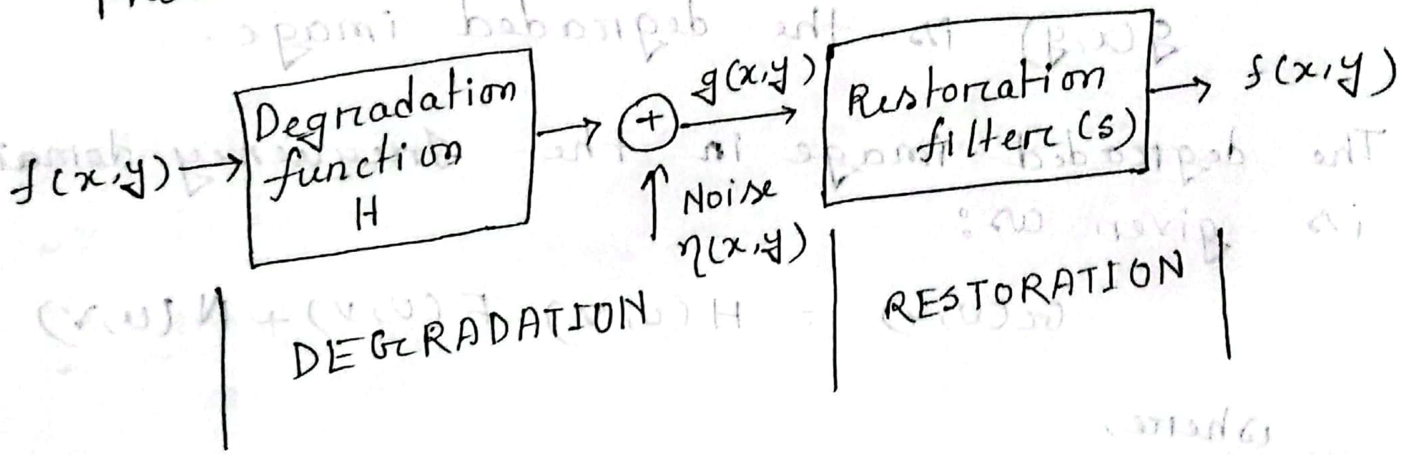
→ It is similar to image similar to image enhancement but more objective.

Restoration attempts to reconstruct or recover an image that has been degraded by using a prior or estimated knowledge of the degradation phenomenon.

Restoration process involves:

- modeling the degradation.
- applying the inverse process in order to recover the original image.

A model of the image Degradation / Restoration process →



→ The objective of restoration is to obtain an estimate $\hat{f}(x,y)$ of the original image $f(x,y)$

→ We want the estimate to be as close as possible to the original input image and in general, the more we know about H and η the closer $\hat{f}(x,y)$ will be to $f(x,y)$

The degraded image in the spatial domain is given as:

$$g(x,y) = h(x,y) \times f(x,y) + \eta(x,y)$$

where,

$h(x,y)$ is the spatial representation of the degradation function.

$\eta(x,y)$ is the additive noise term.

$g(x,y)$ is the degraded image.

The degraded image in the frequency domain is given as:

$$G(U,V) = H(U,V)\, F(U,V) + N(u,v)$$

where,

$H(U,V)$ is the Fourier Transform of $h(x,y)$

$N(u,v)$ is the Fourier Transform of $\eta(x,y)$

$G(U,V)$ is the Fourier Transform of $g(x,y)$

$F(u,v)$ is the Fourier Transform of $f(x,y)$

→ Some restoration techniques are best formulated in the spatial domain, while others are best suited for the frequency domain.

→ For example, spatial processing is applicable when the only degradation is additive noise.

→ on the other hand, degradation such as image blur is best handled in the frequency domain.

→ While dealing with degradation only due to noise we assume that H is the identity operator.

Noise Models

* Sources of noise

→ Image aquisition, digitization, transmission

→ Image sensors: noise occurs due to environmental conditions and by the quality of the sensing elements.

* White noise

→ In signal processing, white noise is a random signal having equal intensity at difficult frequencies giving it a constant power spectral density.

→ The Fourier spectrum of noise is constant

Types of noise

1. Gaussian Noise

→ Random noise that enters a system can be modelled as a Gaussian or normal distribution.

→ The noise affects both dark and light areas of an image

→ The PDF of a Gaussian random variable, z, is given by,

$$P(z) = \frac{1}{\sqrt{2\pi}a} e^{-(z-\mu)^2/2a^2}$$

where, $z \rightarrow$ gray level

$\mu \rightarrow$ mean of average values of z

$a \rightarrow$ standard derivation

$a^2 \rightarrow$ variance

2. Impulse noise:

→ It is also known as shot noise, solt and pepper noise and binary noise.

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

→ The PDF of (bipolar) impulse noise is given by,

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

The either P_a or P_b is zero, impulse impulse noise is called unipolar.

3. Poisson noise

→ This type of noise manifests as a random structure ~~are~~ in images.

→ It is very common in X-ray images

→ The PDF of poisson noise is given by:

$$p(z) = \frac{(np)^z e^{-np}}{z!}$$

where, n → total no. of pixels

z → gray level

p → ratio of noise pixels to the total no. of pixels.

4. Exponential noise

→ This type of noise occurs mostly due to the illumination problem.

→ It is observed in laser imaging

→ The PDF of exponential noise is given by:

$$p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $a > 0$

Mean → $1/a$

variance → $\frac{1}{a^2}$

5. Gamma noise.

→ This type of noise also occurs mostly due to the illumination problems.

→ The PDF is given by

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

where $a > 0$ and b is a positive integer.

Mean → b/a a → min gray value

variance → b/a^2 b → max " "

6. Rayleigh noise

→ This type of noise is mostly present in range images.

→ Range images are mostly used in remote sensing application.

→ The PDF is given by :

$$P(Z) = \begin{cases} \frac{2}{b}(Z-a)\, e^{-\frac{(Z-a)^2}{b}} & \text{for } Z \geq a \\ 0 & \text{otherwise} \end{cases}$$

Mean → $a + \sqrt{\pi b/4}$

variance → $\frac{b(4-\pi)}{4}$

7. Uniform noise

→ It is also a very popular noise which occurs in images where different values of noise are equally probable.

→ It occurs because of quantization noise

→ The PDF is given by :

$$P(Z) = \begin{cases} 1/b-a & \text{for } a \leq f(x,y) \leq b \\ 0 & \text{otherwise} \end{cases}$$

Mean → $a+b/2$

variance → $\frac{(b-a)^2}{12}$

"Image Morphological Processing"

* In digital Image processing, Morphology are mathematical morphology is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and convex hull.

Morphological operations can be used to remove imperfections in the image and provide information on the form and structure of the image.

→ used both ~~preposssing~~ preprocessing & post processing, morphological filtering, thinning & pruning.

→ The language of mathematical morphology is set theory.
↳ represent object in image.

→ Binary image → 2D integer space (Z^2)

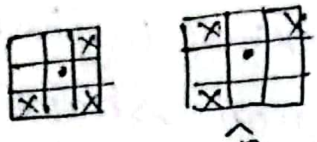
→ Grayscale image → Z^3

→ two types of set of pixels:

- objects (set of foreground pixel)
- structuring elements (set of both foreground & Background pixel)

The reflection of a set (structuring element) B about its origin, denoted by $\hat{B}$ -

$\hat{B} = \{ w \mid w = -b, \text{ for } b \in B \}$ $\quad B(x,y) \rightarrow \hat{B}(-x,-y)$



→ Fig: a

B $\quad$ $\hat{B}$

That is, if B is a set of points in 2-D, then $\hat{B}$ is the set of points in B whose (x,y) coordinates have been replaced by $(-x,-y)$.

Fig: (a) shows several examples of digital set and their reflection. The dot denotes the origin of the SE.

The translation of a set B by point $z = (z_1, z_2)$, denoted $(B)_z$, is defined as

$(B)_z = \{ c \mid c = b+z, \text{ for } b \in B \}$

That is, if B is a set of pixels in 2-D, then $(B)_z$ is the set of pixels in B whose (x,y) coordinates have been replaced by $(x+z_1, y+z_2)$.

This construct is used to translate (slide) a structuring element over an image and each location perform a set operation between the structuring element and the area of the image directly under it.

There is a slight overlap between Morphology & image segmentation.

once segmentation is complete, morphological operation can be used to remove imperfections in the segmented image and deliver information on the shape and structure of the image.

⇒ Structuring element : It is a matrix or a small sized template that is used to traverse an image. The structuring element is positioned at all possible locations in the image, and it is compared with the connected pixels, It can be of any shape.

⇒ Fit : When all the pixels in the structuring element covers the pixels of the object, we call it fit.

⇒ Miss : When no pixels in the structuring element cover the pixels of the object.

⇒ Hit : When at least one of the pixels in the structuring element covers the pixels of the object.

Erosion :

Morplological expressions are written in terms of structuring elements and a set, A of foreground pixels, or in terms of structuring elements and an image, I, that contain A.

We consider the former approach first. With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as,

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}$$

where, A = set of foreground pixels

B = structuring element.

z = z's are foreground values (1's)

this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

Erosion shrinks-ens the image pixels, or removes pixels on object boundaries. First, we traverse the structuring element over the image object to perform an erosion operation.

The value of the output pixel is the minimum value of all the pixels in the neighborhood. A pixel is set to 0(zero) if any of the neighborhood pixels have the value zero (0).

pixel (output) = 1 : If Fit

pixel (output) = 0 ; If otherwise.

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{matrix} \quad \rightarrow \quad \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

structuring Element   input   output

Approach:

- Read the RGB image
- Using function im2bw(), convert the RGB image to a binary image
- Create a structuring element.
- Store the number of rows and column in an array and loop through it
- Create a zero matrix of the same size as the size of real image.
- Leaving the boundary pixels start moving the structuring element on the image and

start comparing the pixel with the pixels present in the neighborhood.

- If the value of the neighborhood pixel is zero, then change the value of that pixel to zero.

Application:

- Noise Reduction
- object segmentation.
- shape analysis
- Feature extraction.

Dilation:

For sets A and B is Z^2 (Binary image), dilation of A by B is denoted by $A \oplus B$

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \phi \}$$

$z \mid (B)_z \cap A \neq \phi$

- Dilation expands the image pixels i.e. it is used for expanding an element A by using structuring element B.

- Dilation adds pixels to object boundaries.

- The value of the output pixel is the maximum value of all the pixel in the neighborhood. A pixel is set to 1 if any of the neighborhood pixels have the value 1.

pixel (output) = 1 ; if Hit

pixel (output) = 0 ; otherwise.

Approach:

- Read the RGB image
- Using function im2bw(), convert the RGB image to binary image.
- Create a structuring element
- Store the number of rows and columns in an array and loop through it.
- Create a zero matrix of the size same as the size of real image.
- Leaving the boundary pixels start moving the structuring element on the image and start comparing the pixel with the pixels present in the neighborhood.
- If the value of the neighborhood pixel is 1, the change the value of that pixel to 1

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

Application:

- object filling
- Noise reduction
- Boundary thickening
- Test enhancement
- repair break & instrusions
- Resolution/ intesity increases.

	Dilation	Erosion
1.	It increases the size of product	It decreases the size of product
2.	It fills the holes and broken area	It removes the smale anomalies
3.	It increases the brightness of the object	It removes the objects smaller than the structuring element
4.	It is XOR of A and B	It is dual of dilation.

5. It is used priore in Closing operation.	It is used latere in closing operation
6. It is used latere in opening operation.	It is priore in oppening operation.
7. Distributed, duality, translation & decomposition properties are followed	It also follows the different properties like duality etc.
8. It connects the areas that are separated by space smaller that structuring element.	It reduces the brightness of the bright objects

Compound operations –

closing (first dilation → erosion)

$(A \oplus B) \ominus B$

opening (first erosion → dilation)

$(A \ominus B) \oplus B$

Opening: It can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation.

Opening is an idempotent operation: once an image has been opened, subsequent openings with the same structuring element have no further effect on that image.

The opening of set A by structuring element B, denoted by $A \circ B$, is defined as

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

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B.

A

1	1	1	0	1	1	1
1	1	1	1	1	1	1
1	1	1	0	1	1	1

B

1	1	1
1	1	1
1	1	1

$$A \ominus B = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$A \circ B = (A \ominus B) \oplus B = \begin{matrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{matrix}$$

Closing: closing generally smooth of contours, but, ~~as opposed to opening~~ while It can fill holes in the regions sizes. It generakeeping the initial region sizes. It generally fuses narrow breaks and long thin gulfs, eliminates smalls holes and fills gaps in the contour.

The closing of set A by structuring element B, denoted $A \cdot B$, is defined as

$$A \cdot B = (A \oplus B) \ominus B$$

The closing of A by B is simply the dilation of A by B, followed by erosion of the result by B.

$$A = \begin{array}{ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{array} \qquad B = \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}$$

$A \oplus B$

$$\begin{array}{ccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$A \cdot B = (A \oplus B) \ominus B$

$$\begin{array}{ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

Hit & Miss transform:

It is perform in much the same way as other
morphological operators, by translating the
origin of the structuring element to all points
in the image, and then comparing the struct
uring element the understanding image pixels.
If the foreground and background pixels in
structuring element exactly match foreground
& background pixel in the image then the
pixel underneath the origin of the structur-
ing element is set to the foreground color,
If it doesn't match, then that pixel is set to
the background color.

EX:

	1	
0	1	1
0	0	

Basic Morphological Algorithm:

The principle application of morphology is extracting image components that are useful in the representation and description of shape. Morphological algorithm are used for boundaries extraction, skeletonization and thinning.

Boundary Extraction:

fig

The boundary of a set A of foreground pixels, denoted by $\beta(A)$, can be obtained by first eroding A by a suitable structuring element B, and then performing the set difference between A and its erosion. That is,

$$\beta(A) = A - (A \ominus B)$$

$\beta(A) = A - (A \ominus B)$

$\beta(A) = A - \ldots$

$$\underset{A}{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{matrix}} \qquad \underset{B}{\begin{matrix} 1 \\ 1 \\ 1 \end{matrix}} \rightarrow \underset{A \ominus B}{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}} \rightarrow \underset{A - A \ominus B}{\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix}}$$

Hole Filling :

A hole may be defined as a background region surrounding by a connected border of fore-ground pixels. In this section, we develop an algorithm based on set dilation, complementation, and intersection for filling holes in an image.

Let, A denote a set whose elements are 8-connected boundaries, with each boundary enclosing a background region. Given a point in each hole, the objective is to fill all the holes with foreground elements (1's).

We begin by forming an array, X_0, of 0's except at locations in X_0 that correspond to pixels that are known to be holes, which we set to 1. Then the following procedure fills all the holes with 1's.

$$X_K = (X_{K-1} \oplus B) \cap I^C \qquad K = 1, 2, 3 \ldots$$

A.

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

$X_0 \leftarrow \boxed{0}$

B

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

A^C

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

Image Segmentation

↓

based on image intensity value

(boundary approach) ← → discontinuty (to partition an image into region based on abrupt changes in intensity)

→ similarity

↓

(region approach) (partitioning an image into regions that are similar according to predefined crieteria)

Segmentation is the process of partitioning a digital image into multiple regions and extracting the meaningful region which is known as Region of interest (ROI).

The purpose of image segmentation is to simplify the representation of an image, making it easier to analyze and extract meaningful information from it. This technique is widely used in computer vision and image processing for various application such as object detection, recognition, tracking medical image analysis and more

⇒ Let R is the entire region & R into subregions $R_1, R_2, \ldots R_n$ such that

a) $\bigcup_{i=1}^{n} R_i = R$

b) R_i is connected set, for $i = 0, 1, 2, \ldots, n$

c) $R_i \cap R_j = \phi$. if $i \neq j$

d) $P(R_i) =$ True for $i = 1, 2, \ldots n$

e) $P(R_i \cup R_j) =$ False for any adjacent regions R_i and R_j.

Segmentation Algorithm

- Based on User Interaction
  - Manual
  - Automatic
  - Semi-Automatic
- Based on pixel Relation-ship
  - contextual (region based or global)
  - Non-contextual (pixel based or local)

Image Segmentation Technique

- Discontinuity
  - Edge-based (point, line & Edge)
    - Gradient
      - Sobel
      - Robert
      - Prewitt
    - Gaussian based
      - Canny edge detection
      - Laplacian of Gaussian
- Similarity
  - clustering based
  - Thresholding
    - Global
    - Local
  - Region based
    - Region growing
    - Region Splitting & Merging

Detection of Discontinuity :

→ three basic types of gray level discontinuities (point, lines, edges)

→ Use the image sharpening technique

- The first order derivatives produce thicker edges
- The second order derivatives (Laplacian operation) have a strong response to find detail, such as thin lines and isolated points and noise.

→ can be done by running a mask through the image

$$R = W_1 Z_1 + W_2 Z_2 + \cdots W_9 Z_9$$
$$= \sum_{K=1}^{9} W_K Z_K$$

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

point Detection :

→ Apply Laplacian filter to the image to obtain $R(x,y)$

→ Create binary image by threshold.

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & ;\text{ otherwise} \end{cases}$$

T is a non-negative threshold

-1	-1	-1
-1	8	-1
-1	-1	-1

This formulation simply measures the weighted difference between a pixel and its 8-neighbors. Intuitively, the idea is that the intensity of an isolated point will be quite different from its surroundings and thus will be easily detectable by this type of kernel.

Ex :- Input image Kernel

1	3	2
0	1	5
6	8	2

×

1	1	1
1	-8	1
1	1	1

for example threshold value = 6

output image

0	0	0
0	0	0
1	1	0

2. Line Detection :

When a discontinuity takes place through out the image which can be present in horizontal +45°, vertical or -45° of pattern which is present in an line format the change in the pixel value of the image that making a line and make it different from the image is called as a line detection in an image.

* We use second order derivatives which result in a stronger filters response and produce thinner lines than first derivatives.

Horizontal

-1	-1	-1
2	2	2
-1	-1	-1

+45°

2	-1	-1
-1	2	-1
-1	-1	2

Vertical

-1	2	-1
-1	2	-1
-1	2	-1

-45°

-1	-1	2
-1	2	-1
2	-1	-1

3. Edge detection:

→ Edge detection method is mostly used to finding object boundaries in images.

→ It works by detecting discontinuities is brightness.

→ Edge detection is used for image segmentation and data extraction in areas such as image processing, computer vision and machine vision

→ Common edge detection algorithms include Sobel, canny, prewitt, Roberts and Fuzzy logic Methods.