Syllabus:

Exam	TT-1	TT-2	AVG	Term Work	Oral/Practical	End of Exam	Total
Marks	20	20	20	25	25	80	150

#	Module	Details Contents	No.
1.	Discrete-Time Signal and Discrete-Time System	▪ Introduction to Digital Signal Processing, Sampling and Reconstruction, Standard DT Signals, Concept of Digital Frequency, Representation of DT signal using Standard DT Signals, Signal Manipulations(shifting, reversal, scaling, addition, multiplication) ▪ Classification of Discrete-Time Signals, Classification of Discrete-Systems ▪ Linear Convolution formulation for 1-D and 2-D signal (without mathematical proof), Circular Convolution (without mathematical proof), Linear convolution using Circular Convolution. Auto and Cross Correlation formula evaluation, LTI system, Concept of Impulse Response and Step Response, Output of DT system using Time Domain Linear Convolution.	01
2.	Discrete Fourier Transform	▪ Introduction to DTFT, DFT, Relation between DFT and DTFT, IDFT ▪ Properties of DFT without mathematical proof (Scaling and Linearity, Periodicity, Time Shift and Frequency Shift, Time Reversal, Convolution Property and Parsevals' Energy Theorem). DFT computation using DFT properties. ▪ Transfer function of DT System in frequency domain using DFT. Linear and Circular Convolution using DFT, Convolution of long sequences, Introduction to 2-D DFT	16
3.	Fast Fourier Transform	▪ Need of FFT, Radix-2 DIT-FFT algorithm ▪ DIT-FFT Flow graph for N=4 and 8, Inverse FFT algorithm ▪ Spectral Analysis using FFT	21
4.	Digital Image Fundamentals	▪ Introduction to Digital Image, Digital Image Processing System, Sampling and Quantization ▪ Representation of Digital Image, Connectivity ▪ Image File Formats: BMP, TIFF and JPEG.	29
5.	Image Enhancement in Spatial domain	▪ Gray Level Transformations, Zero Memory Point Operations ▪ Histogram Processing, Histogram equalization. ▪ Neighborhood Processing, Spatial Filtering, Smoothing and Sharpening Filters, Median Filter.	35
6.	Image Segmentation	▪ Segmentation based on Discontinuities (point, Line, Edge) ▪ Image Edge detection using Robert, Sobel, Previtt masks, Image Edge detection using Laplacian Mask.	49

We have tried to cover almost every important topic from above syllabus (theory only). If you feel something is miss out and not solved in this solution, then do contact us via WhatsApp (+919930038388) or Facebook (FB/BackkBenchersCommunity). We will provide solution for the same.

For sums, refer Last Moment Tuitions (LMT) Videos. LMT is a YouTube Channel which provide the clear knowledge of various subjects of Engineering.

CHAP - 1: DISCRETE-TIME SIGNAL AND DISCRETE-TIME SYSTEM

Q1. Explain Digital Signal Processing

Ans: **[P | Medium]**

DIGITAL SIGNAL PROCESSING:

1. Digital Signal Processing (DSP) is the subfield of **Signal Processing**.
2. Digital signal processing is the processing of signals by digital means.
3. Digital signal processing is the use of digital processing, such as by computers, to perform a wide variety of signal processing operations.
4. DSP is defined as changing or analyzing information which is measured as discrete sequences of numbers.
5. DSP is used in a **wide variety of applications**.
6. Figure 1.1 shows the basic elements of DSP and its requirement.

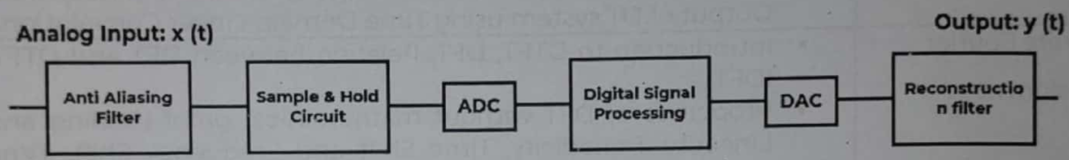


Figure 1.1: Basic Elements of DSP.

The elements of DSP are as follows:

I) Input signal:

1. It is the signal generated from some transducer or from some communication system.
2. It may be biomedical signal like **ECG or EEG**.
3. It is analog in nature.
4. It is denoted by x (t).

II) Anti-aliasing fitter:

1. Anti-aliasing filter is basically a **low pass filter**.
2. It avoids aliasing effect.
3. It removes the high frequency noise contain in input signal.

III) Sample and hold circuit:

1. This block takes the samples of input signal.
2. It keeps the voltage level of input signal relatively constant which is the requirement of ADC.

IV) Analog to digital converter (ADC):

1. This block is used to convert **analog signal into digital form**.

V) Digital signal processor:

1. DSP is used to process input signal **digitally**.
2. DSP processors such as ADSP 2100 or TMS 320 are basically used.

VI) Digital to analog converter (DAC):

1. The output of digital signal processor is digital in nature.

Digitals signals can be easily stored.	It is difficult to solve analog signals.
Mathematical processing algorithm can be easily implemented.	Mathematical processing algorithm is difficult to implement.

Q3. Explain Signal Manipulations

Ans: **[P | Medium]**

SIGNAL MANIPULATIONS:

1. Signals can be composed by manipulating and combining other signals.
2. Manipulations are generally compositions of a few basic signal transformations.
3. These transformations may be classified either as those that are transformations of the independent variable n or those that are transformations of the amplitude of x(n) i.e. dependent variable.

TRANSFORMATION OF THE INDEPENDENT VARIABLE:

1. Sequences are often altered and manipulated by modifying the index n as follows:
   $Y(n) = x(f(n))$
2. Where f (n) is some function of n.
3. If, for some value of n, f (n) is not an integer, y(n) = x(f(n)) is undefined.
4. Determining the effect of modifying the index n may always be accomplished using a simple tabular approach of listing, for each value of n, the value of f (n) and then setting y(n) = x(f (n)).
5. However, for many index transformations this is not necessary, and the sequence may be determined or plotted directly.
6. The most common transformations include shifting, reversal, and scaling, which are defined below.

I) Shifting:

1. This is the transformation defined by $f(n) = n - n_0$.
2. Consider $y(n) = x(n - n_0)$
3. If n_0 is positive, x(n) is shifted to the right by n_0 samples (this is referred to as a delay)
4. If n_0 is negative, x(n) is shifted to the left by n_0 samples (referred to as an advance).

II) Reversal:

1. This is the transformation defined by $f(n) = -n_0$ and simply involves "flipping" the signal x(n) with respect to the index n.

III) Time Scaling:

1. This transformation is defined by f (n) = Mn or f (n) = n/N, where M and N are positive integers.
2. In the case of f(n) = Mn, the sequence x(Mn) is formed by taking every M^{th} sample of x(n) (this operation is known as down-sampling).
3. With f (n) = n/N the sequence y(n) = x(f (n)) is defined as follows:

$$y(n) = \begin{cases} x\left(\frac{n}{N}\right) & n = 0, \pm N, \pm 2N, \cdots \\ 0 \end{cases}$$

4. Otherwise, this operation is known as up-sampling.
5. Examples of shifting, reversing & time scaling a signal are illustrated in figure 1.2.

2. But the required final output is analog in nature.
3. So to convert digital signal into analog signal DAC is used.

VII) Reconstruction filter:

1. Output signal of DAC is analog, which means it is a continuous signal.
2. But it may contain high frequency components.
3. Such high frequency components are unwanted.
4. To remove these components; **reconstruction filter is used**.

ADVANTAGES:

1. It is cheaper to implement.
2. It can be stored on disk.
3. It can be easily duplicated.
4. It is can be easily upgraded as compared to analog system.
5. Small size.

DISADVANTAGES:

1. DSP hardware is more expensive than general purpose microprocessors & micro controllers.
2. Finite word length problems.
3. Bandwidth Limitations.
4. Power consumption is high.

APPLICATIONS:

1. Filtering.
2. Image Processing.
3. Speech Recognition.
4. Signal Analysis.
5. Wave Form Generation.

Q2. Comparison between Digital and Analog Signal Processing

Ans:

[P | Medium]

Table 1.1 shows the comparison between Digital and Analog Signal Processing.

Digital Signal Processing	Analog Signal Processing
Digital Systems can be easily replicated.	It is tedious to replicate analog systems.
It is more versatile.	It is less versatile.
It has more system complexity.	It has less system complexity.
Software up-gradation can be done easily.	Software up-gradation is difficult.
It has better accuracy.	Accuracy is less.
Universal compatibility is possible in DSP.	Universal compatibility is not possible in ASP.
It requires more power consumption.	It requires less power consumption.
It is more flexible.	It is less flexible.

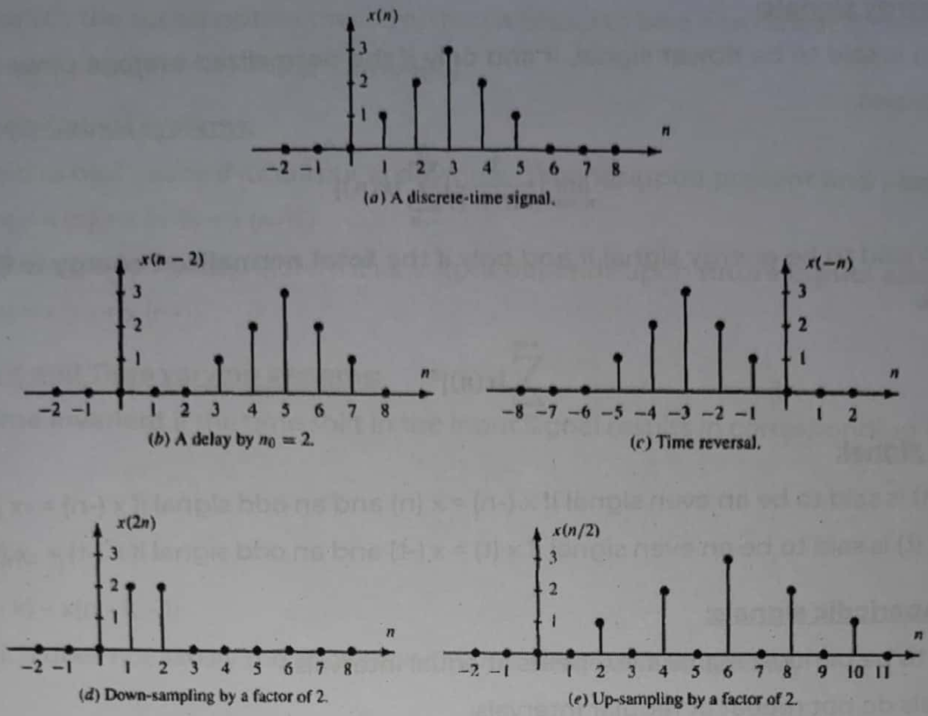


Figure 1.2: Examples of shifting, reversing & time scaling

ADDITION, MULTIPLICATION, AND SCALING:

1. The most common types of amplitude transformations are addition, multiplication, and scaling.
2. Performing these operations is straightforward and involves only point wise operations on the signal.
3. **Addition:** The sum of two signals:

   $y(n) = x_1(n) + x_2(n)$ Where $\infty < n < \infty$ is formed by the point wise addition of the signal values.
4. **Multiplication**: The multiplication of two signals:

   $Y(n) = x_1(n)\, x_2(n)$ Where $\infty < n < \infty$ is formed by the point wise product of the signal values.
5. **Scaling:** Amplitude scaling of a signal x(n) by a constant c is accomplished by multiplying every signal value by c:

   $y(n) = cx(n)$ Where $\infty < n < \infty$ This operation may also be considered to be the product of two signals. x(n) and f (n) = c.

Q4. Explain classification of continuous & discrete time signals

Ans: **[P | Medium]**

CLASSIFICATION OF CONTINUOUS & DISCRETE TIME SIGNALS:

I) Deterministic and Random signals:

1. A deterministic signal is one which can be completely represented by Mathematical equation at any time.
2. In a deterministic signal there is no uncertainty with respect to its value at any time.
3. E.g.: x (t) = coswt & x (n)=2πft
4. A random signal is one which **cannot be represented by any mathematical equation**.
5. E.g.: Noise generated in electronic components, transmission channels, cables etc.

II) Power and Energy signals:

1. The signal x (n) is said to be power signal, if and only if the normalized average power p is finite and non-zero. I.e. 0<p<4

$$P = \lim_{n\to\infty}\left(\frac{1}{2N+1}\right)\sum_{-N}^{N}|x(n)|^{2}$$

2. A signal x (n) is said to be energy signal if and only if the **total normalized energy is finite and non-zero.** I.e. 0<E< 4

$$\sum_{-\infty}^{+\infty}|x(n)|^2$$

III) Odd and even signal:

1. A DT signal x (n) is said to be an even signal if x (-n) = x (n) and an odd signal if x (-n) = -x (n).
2. A CT signal is x (t) is said to be an even signal if x (t) = x (-t) and an odd signal if x (-t) = -x (t).

IV) Periodic and Aperiodic signals:

1. A signal is said to be periodic signal if it repeats at equal intervals.
2. Aperiodic signals do not repeat at regular intervals.
3. A CT signal which satisfies the equation x (t) = x (t+T0) is said to be periodic
4. DT signal which satisfies the equation x (n) = x (n + N) is said to be periodic.

Q5. Define System?

Ans: **[P | Medium]**

SYSTEM:

1. A system is a set of **elements or functional blocks** that are connected together and produces an output in response to an input signal.
2. E.g.: An audio amplifier, attenuator, TV set etc.

CLASSIFICATION OR CHARACTERISTICS OF CT AND DT SYSTEMS:

I) Static and Dynamic system:

1. A system is said to be static or memory less if its output depends upon the present input only.
2. Static System has no storage devices.
3. **Example:** y (n) = 4x (n) & y (n) = log x (n)
4. The system is said to be dynamic with memory if its output depends upon the present and past input values.
5. Dynamic System has the storage devices.
6. **Example:** y (n) = x (n) + x (n-2) & y (n) = x (n) * x (n-1)

II) Linear and Non-linear systems:

1. A system is said to be linear if superposition theorem applies to that system.
2. Superposition principle states that the response of the system to the weighted sum of signal is equal to corresponding weighted sum of output of a system to individual input signal.
3. **Example:** $H[a_1x_1(n) + a_2x_2(n)] = a_1H[x_1(n)] + a_2H[x_2(n)]$

4. If it does not satisfy the superposition theorem, then it is said to be a non- linear system.
5. **Example:** $H[a_1x_1(n) + a_2x_2(n)] \; != a_1H[x_1(n)] + a_2H[x_2(n)]$

III) Causal and non-Causal systems:

1. A system is said to be a causal if its output at any time depends upon **present and past inputs only**.
2. **Example:** y (n) = x (n) + x (n-2) + x (n-16)
3. A system is said to be non-causal system if its output depends upon **future inputs also**.
4. **Example:** y (n) = x (n) + x (n+1)

IV) Time invariant and Time varying systems:

1. A system is **time invariant** if the time shift in the input signal results in corresponding time shift in the output.
2. **Example:**

   y(n) = x(n) – x(n – 1)

   y(n - k) = x(n - k) – x(n - k – 1)
3. A system which does not satisfy the above condition is **time variant system**.
4. **Example:**

   y(n) = nx(n)

   y(n - k) = (n – k) x (n - k)

   = nx(n – k) – kx(n – k)

   But x(n - k) → nx(n - k) != y(n – k)

V) Stable and Unstable systems:

1. When the system **produces bounded output for bounded input**, then the system is called bounded input, bounded output stable.
2. A system which does not satisfy the above condition is called a unstable system.

Q6. Explain Convolution & Correlation?

Ans: **[P | Medium]**

CONVOLUTION:

1. Convolution is a mathematical way of **combining two signals** to form a third signal.
2. Using the strategy of **impulse decomposition**, systems are described by a signal called the impulse response.
3. Convolution is important because it relates the three signals of interest: **the input signal**, **the output signal**, and **the impulse response**.
4. Convolution provides the **mathematical framework** for Digital Signal Processing.
5. It states that for a Linear Time-Invariant system, if the input sequence x (n) and impulse response h (n) are known, y (n) can be found out from the convolution sum.
6. The sum is represented as:

   y (n) = x (n) * h (n). Where * denotes the convolution operation.

Properties:

1. **Commutative Law:**
   a. The commutative property states that the order in which two sequences are convolved is not important.
   b. Mathematically, the commutative property is $x(n) * h(n) = h(n) * x(n)$
   c. From a systems point of view, this property states that a system with a unit sample response $h(n)$ and input $x(n)$ behaves in exactly the same way as a system with unit sample response $x(n)$ and an input $h(n)$.
   d. This is illustrated in Figure 1.3

Figure 1.3: Commutative Law

2. **Associative Law:**
   a. The convolution operator satisfies the associative property, which is

$$\{x(n) * h_1(n)\} * h_2(n) = x(n) * \{h_1(n) * h_2(n)\}$$

   b. From a systems point of view, the associative property states that if two systems with unit sample responses $h_1(n)$ and $h_2(n)$ are connected in cascade as shown in Figure 1.4, an equivalent system is one that has a unit sample response equal to the convolution of $h_1(n)$ and $h_2(n)$:

$$h_{eq}(n) = h_1(n) * h_2(n)$$

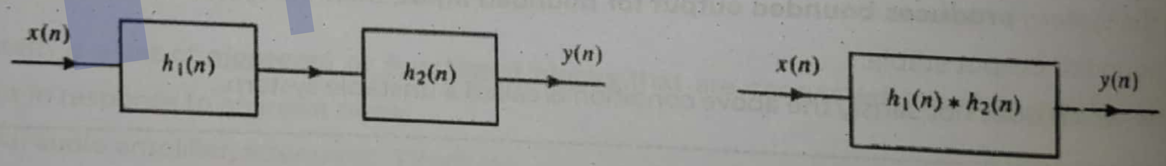


Figure 1.4: Associative Law

3. **Distributive Law:**
   a. The distributive property of the convolution operator states that

$$x(n) * \{h_1(n) + h_2(n)\} = x(n) * h_1(n) + x(n) * h_2(n)$$

   b. From a systems point of view, this property asserts that if two systems with unit sample responses $h_1(n)$ and $h_2(n)$ are connected in parallel, as illustrated in Figure 1.5, an equivalent system is one that has a unit sample response equal to the sum of $h_1(n)$ and $h_2(n)$:

$$h_{eq}(n) = h_1(n) + h_2(n)$$

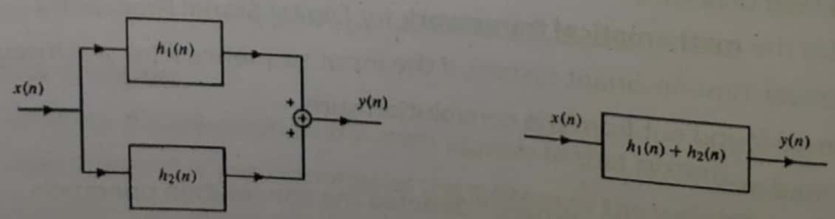


Figure 1.5: Distributive Law

CORRELATION:

1. It is used to **compare two signals.**
2. It is very important operation in signal & image processing.
3. Correlation is the **measure of degree** to which two signals are similar.
4. Correlation as a mathematical operation **closely resembles convolution.**
5. Correlation of two separate signals is known as **cross correlation.**
6. While correlation of signal with itself is known as **auto correlation.**

Auto - correlation:

$$Rxy\ (l) = \sum_{-\infty}^{+\infty} x\,(n).\,y\,(n-l)$$

Cross Correlation:

$$Rxx\ (l) = \sum_{-\infty}^{+\infty} x\,(n).\,x\,(n-l)$$

Properties:

1. Auto correlation is an even function.
2. The cross correlation is not commutive.
3. The result of auto correlation is maximum when signal matches with itself and there is no phase shifting.

Q7. LTI System

Ans: **[P | Medium]**

LTI SYSTEM:

1. LTI stands for **Linear Time Invariant Systems.**
2. LTI Systems are a class of systems used in signals and systems that are both linear and time-invariant.
3. Linear systems are systems whose outputs for a linear combination of inputs are the same as a linear combination of individual responses to those inputs.
4. Time-invariant systems are systems where the output does not depend on when an input was applied.
5. These properties make LTI systems easy to represent and understand graphically.
6. LTI systems are the most important class of systems because the behavior of the system is known by knowing its response to unit sample input (x(n)).

Causality of LTI System:

$$y(n) = \sum_{-\infty}^{\infty} h(k)x(n-k) = \sum_{-\infty}^{-1} h(k)x(n-k) + \sum_{0}^{\infty} h(k)x(n-k)$$

$$= \underbrace{\left[\ldots + h(-2)x(n+2) + h(-1)x(n+1)\right]}_{\text{this depends on future values of } x(n)} + \left[h(0)x(n) + h(1)x(n-1) + \ldots\right]$$

Hence, for a system to be casual, its h(n) must be zero for n < 0.

BIBO Stability of LTI System:

1. BIBO Stands for Bounded Input Bounded Output.
2. It is an important and generally desirable system characteristic.
3. A system is BIBO stable if every bounded input signal results in a bounded output signal, where boundedness is the property that the absolute value of a signal does not exceed some finite constant.
4. In terms of time domain features, a discrete time system is BIBO stable if and only if its impulse response is absolutely summable.
5. Equivalently, in terms of z-domain features, a continuous time system is BIBO stable if and only if the region of convergence of the transfer function includes the unit circle.

Discrete Time BIBO Stability:

A bounded signal is any signal such that there exists a value such that the absolute value of the signal is never greater than some value. Since this value is arbitrary, what we mean is that at no point can the signal tend to infinity, including the end behavior.

Figure 1.6: BIBO Stability

Time Domain Conditions:

The continuous-time LTI system with impulse response h(n) is BIBO stable **if and only if** it is **absolutely summable**. That is:

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

Q8. Differentiate IIR & FIR Systems

Ans:

[P | Medium]

Table 1.2: Comparison between IIR & FIR System.

IIR	FIR
IIR system are recursive.	FIR system are non-recursive.
It requires less memory.	It requires more memory.
It has lower computational complexity.	It has higher computational complexity.
IIR filters do not have linear phase characteristics.	FIR filters provides linear phase.
Implementation involves less parameters.	Implementation involves more parameters.

Impulse response h(n) is infinite.	Impulse response h(n) is finite.
For IIR filters, each output must be individually calculated.	FIR filters are suited to multi-rate applications. It provides computation efficiency.
It is more susceptible to problems of finite length arithmetic.	It is less susceptible to problems of finite length arithmetic.
IIR filter have both numerators and denominators.	FIR filters have only numerators.
Stability depends on system design.	Stability is always guaranteed.

Q9. Sampling & Quantisation

Ans: **[P | Medium]**

SAMPLING:

1. Sampling is defined as "The process of measuring the instantaneous values of continuous-time signal in a discrete form".
2. Sample is a piece of data taken from the whole data which is continuous in the time domain.
3. When a source generates an analog signal and if that has to be digitized, having 1s and 0s i.e., High or Low, the signal has to be discretized in time.
4. This discretization of analog signal is called as **Sampling**.
5. The following figure indicates a continuous-time signal x(t) and a sampled signal $x_s(t)$.
6. When x(t) is multiplied by a periodic impulse train, the sampled signal $x_s(t)$ is obtained.
7. Figure 1.7 shows the example of sampling.

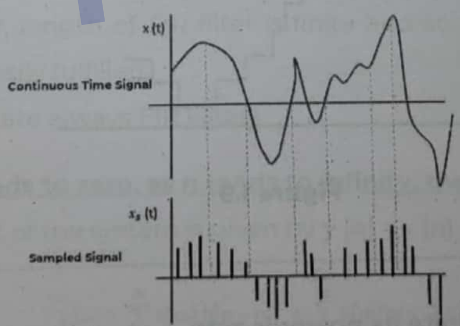


Figure 1.7: Sampling

QUANTISATION:

1. The digitization of analog signals involves the rounding off of the values which are approximately equal to the analog values.
2. The method of sampling chooses a few points on the analog signal and then these points are joined to round off the value to a near stabilized value.
3. Such a process is called as **Quantization**.
4. The following figure 1.8 shows how an analog signal gets quantized.
5. The blue line represents analog signal while the red one represents the quantized signal.

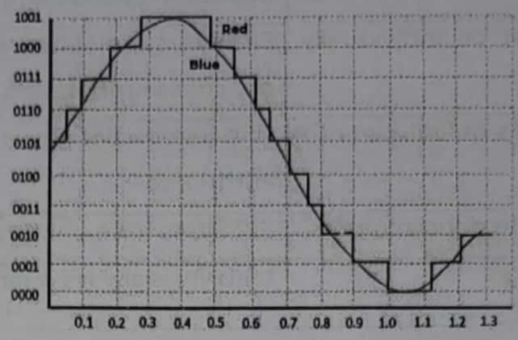


Figure 1.8

6. Both sampling and quantization result in the loss of information.
7. The quality of a Quantizer output depends upon the number of quantization levels used.
8. The discrete amplitudes of the quantized output are called as representation levels or reconstructio levels.
9. The spacing between the two adjacent representation levels is called a quantum or step-size.
10. The following figure 1.9 shows the resultant quantized signal which is the digital form for the give analog signal.

Figure 1.9

Q10. If the energy of the signal is finite its power is zero

Ans: **[P | Medium]**

1. A signal can be categorized into energy signal or power signal.
2. An energy signal has a finite energy, $0 < E < \infty$.
3. In other words, energy signals have values only in the limited time duration.
4. For example, a signal having only one square pulse is energy signal.
5. A signal that decays exponentially has finite energy, so, it is also an energy signal.
6. The power of an energy signal is 0, because of dividing finite energy by infinite time (or length).

$$E = \sum_{n=-\infty}^{+\infty} |x(n)|^2$$

7. On the contrary, the power signal is not limited in time.

. It always exists from beginning to end and it never ends.

. For example, sine wave in infinite length is power signal.

). Since the energy of a power signal is infinite, it has no meaning to us.

. Thus, we use power (energy per given time) for power signal, because the power of power signal is finite, $0 < P < \infty$.

$$P = \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} |x(n)|^2$$

. Hence if the energy of the signal is finite its power is zero.

---

11. State each of the following statement is True OR False. Justify your answer in 4 or 5 sentences.

(a) Linear phase Filters are always IIR

(b) For a causal system h[n] tends to zero, as n tends to infinity, the system is stable.

(c) A stable filter is always causal

(d) A stable, causal FIR filter has its poles lying anywhere inside the unit circle in the z plane

(e) IIR filters have recursive realization always

ns: **[P | Medium]**

) <u>Linear phase Filters are always IIR - FALSE.</u>

For linear phase filter h[n] must be either Symmetric or Antisymmetric.

For symm h[n], h[n] = h[N–1–m] and for antisymmetric h[n], h[n] = – h[N–1–m].

In case of IIR filter this condition is not guaranteed as the length of IIR filter is infinite.

However in case of FIR filter, length of FIR filter is finite and so the condition of symmetric h[n] or antisymmetric h[n] can be easily fulfilled.

Therefore linear phase filters are always FIR Filters.

) <u>For a causal system h[n] tends to zero, as n tends to infinity, the system is stable - TRUE</u>

For causal system the output of the system is given by $y[n] = x[n] * h[n]$

$$y[n] = \sum_{-\infty}^{\infty} x[m]\, h[n-m] = \sum_{m=0}^{\infty} x[m] h[n-m]$$

For bounded input i.e.

$$\sum_{-\infty}^{\infty} |x[m]| < \infty \text{ if } \sum_{n=0}^{\infty} |h[n]| < \infty$$

Then output y[n] is also bounded ie $\Sigma\ |y[n] < \infty$

Therefore system is BIBO stable.

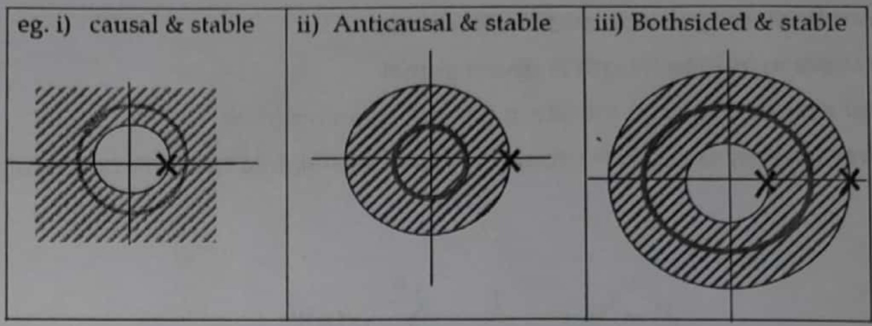
) <u>A stable filter is always causal - FALSE</u>

The condition for stable filter is ROC must include unit circle.

For causal filter, when pole < 1 system is stable.

For Non causal filer with anticausal h[n], when pole > 1 system is stable and with both sided h[n] suppose ROC is $|P_2| > |z| > |P_1|$

Then if $|P_1| < 1$ and $|P_2| > 1$ then system is stable.



(d) A stable, causal FIR filter has its poles lying anywhere inside the unit circle in the z plane – FALSE

1. In case of causal FIR Filter, poles are always only at origin. i.e. pole = 0 .
2. For causal and stable filter all the poles must lie inside the unit circle.
3. Therefore, FIR filters are always stable filter with poles only at origin.

(e) IIR filters have recursive realization always - TRUE

1. In case of IIR Filter, poles can be anywhere in the z plane.
2. Due to pole position, the transfer function H(z) of IIR filter has the form,

$$H(z) = \frac{b_0 + b_1 + \cdots + b_m z}{1 + a_1 + \cdots + a_N z^{-N}}$$

3. This gives difference equation in the form,

$$y[n] = b_0\, x[n] + b_1\, x\,[n-1] + \cdots b_m\, x[n-m] + a_1\, y\,[n-1] + \cdots + a_2\, y[n-N]$$

4. Output of the filter in terms of past output leads to recursive realization.

Q12. Explain Impulse Response & Step Response

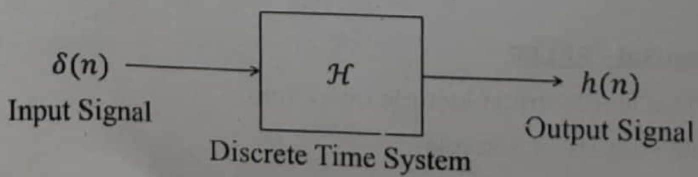
Ans: [P | Medium]

IMPULSE RESPONSE:

1. When the input to a DTS is unit impulse δ(n) then the output is called an impulse response of the system.
2. More generally, an impulse response is the reaction of any dynamic system in response to some external change.
3. It is given as:

$h(n) = H[\delta(n)]$

$= \delta(n) * h(n)$



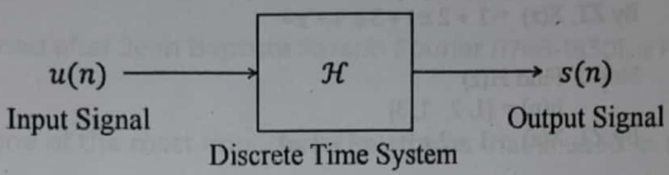
STEP RESPONSE:

1. When the input to a DTS is unit step signal u(n) then the output is called an step response of the system.
2. It is given as:

s(n) = H[u(n)]

s(n) = u(n) * h(n)

h(n) = s(n) – s(n-1)



Q13. Determine cross correlation of the following sequence.

x[n] = {1, 0, 0, 1}, h[n] = {4, 3, 2, 1}

Ans: **[P | Medium]**

Step – 1: To find y[n] for n ≥ 0

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[m\text{-}n] = \sum_{m=0}^{3} x[m]\,h[m\text{-}n]$$

$$y[0] = \sum_{m=0}^{3} x[m]\,h[m] = (1)(4)+(0)(3)+(0)(2)+(1)(1) = \boxed{5}$$

$$y[1] = \sum_{m=0}^{3} x[m]\,h[m\text{-}1] = (1)(.)+(0)(4)+(0)(3)+(1)(2) = \boxed{2}$$

$$y[2] = \sum_{m=0}^{3} x[m]\,h[m\text{-}2] = (1)(.)+(0)(.)+(0)(4)+(1)(3) = \boxed{3}$$

$$y[3] = \sum_{m=0}^{3} x[m]\,h[m\text{-}3] = (1)(.)+(0)(.)+(0)(.)+(1)(4) = \boxed{4}$$

Step – 2: To find y[n] for n < 0

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[m\text{-}n] = \sum_{m=0}^{3} x[m]\,h[m\text{-}n]$$

$$y[-1] = \sum_{m=0}^{3} x[m]\,h[m+1] = (1)(3)+(0)(2)+(0)(.)+(1)(.) = \boxed{3}$$

$$y[-2] = \sum_{m=0}^{3} x[m]\,h[m+2] = (1)(2)+(0)(.)+(0)(.)+(1)(.) = \boxed{2}$$

Q14. The impulse response of LTI system is h(n) = {1, 2, -1, 3}

Determine the output response of the system to input x(n) = {1, 2, 3, 1}

Ans: **[P | Mediu**

To find output y[n] = x[n] * h[n]

By convolution property of ZT, Y(z) = X(z) H(z)

Step-1 Find X(z)

$x(n) = \{1, 2, 3, 1\}$

By ZT, $X(z) = 1 + 2z^{-1} + 3z^{-2} + z^{-3}$

Step-2 Find H(z)

$h(n) = \{1, 2, -1, 3\}$

By ZT, $X(z) = 1 + 2z^{-1} - z^{-2} + 3z^{-3}$

Step-3 Find Y(z)

$Y(z) = X(z)\,H(z)$

$Y(z) = (1 + 2z^{-1} + 3z^{-2} + z^{-3})\,(1 + 2z^{-1} - z^{-2} + 3z^{-3})$

$$
\begin{aligned}
Y(z) = \; & 1 + 2z^{-1} - z^{-2} + 3z^{-3} + \\
& 2z^{-1} + 4z^{-2} - 2z^{-3} + 6z^{-4} + \\
& 3z^{-2} + 6z^{-3} - 3z^{-4} + 9z^{-5} + \\
& z^{-3} + 2z^{-4} - z^{-5} + 3z^{-6}
\end{aligned}
$$

$Y(z) = 1 + 4z^{-1} + 6z^{-2} + 8z^{-3} + 5z^{-4} + 8z^{-5} + 3z^{-6}$

Step-4 Find y[n]

By IZT, $y[n] = \{1, 4, 6, 8, 5, 8, 3\}$ for $n \geq 0$

CHAP - 2: DISCRETE FOURIER TRANSFORM

Q1. **Write short notes on Properties of DFT**

Ans: [P | High]

DISCRETE FOURIER TRANSFORM (DFT):

The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications.

Fourier analysis is named after Jean Baptiste Joseph Fourier (1768-1830), a French mathematician and physicist.

Fourier Transform is one of the most important transforms that is used in Digital Signal Processing & Image Processing.

The Fourier representation of periodic discrete time signals has been extended to non-periodic signals by letting the fundamental period N to infinity.

This Fourier Transform method of representing non-periodic discrete time signals as a function of discrete time frequency, ω is called **Discrete Time Fourier Transform**.

PROPERTIES OF DFT:

Property	Time Domain $x(n)$	Frequency Domain $X(k)$
Scaling	$ax(n)$	$aX(k)$
Linearity	$a_1x_1(n)+a_2x_2(n)$	$a_1X_1(k)+a_2X_2(k)$
Periodicity	$x(n)=x(n+N)$	$X(k)=X(k+N)$
Circular Time Shift	$x((n-m))_N$	$X(k)e^{-j2\pi km/N}$
Circular Frequency Shift	$x(n)e^{j2\pi nm/N}$	$X((k-m))_N$
Time Reversal	$x((-n))_N=x(N-n)$	$X((-k))_N=X(N-k)$
Circular Convolution	$x_1(n) \circledast x_2(n)$	$X_1(k)\,.\,X_2(k)$
Multiplication	$x_1(n)\,.\,x_2(n)$	$\frac{1}{N}[X_1(k) \circledast X_2(k)]$
Parseval's Energy Theorem	$\sum_{n=0}^{N-1}\lvert x(n)\rvert^2$	$\frac{1}{N}\sum_{k=0}^{N-1}\lvert X(k)\rvert^2$
Complex Conjugate	$x^*(n)$	$X^*(N-k)$
Symmetry	$x(n)$ is Real	$X(k)=X^*(N-k)$

Q2. Explain linear convolution using DFT

Ans: [P | High]

LINEAR CONVOLUTION:

1. Multiplication of two sequences in time domain is called as **Linear Convolution**.
2. Linear Convolution is given by the equation y(n) = x(n) * h(n) & calculated as

$$y(n) = \sum_{-\infty}^{\infty} x(k)h(n-k)$$

3. Linear Convolution of two signals returns N-1 elements where N is sum of elements in both sequence

Basic operations:

1. Figure 2.1 shows flow diagram

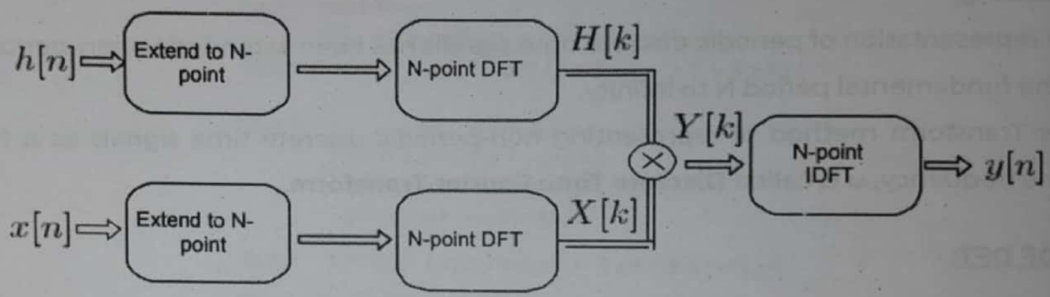


Figure 2.1: Flow Diagram

2. Choose N to be at least (L + P − 1).
3. Pad the two original sequences with zeros to length N.
4. Compute the N-point DFT to obtain H[k] and X[k].
5. Compute the point-wise product: Y [k] = H[k] X[k] Where k = 0, . . .(N − 1).
6. Compute y[n] by taking the N-point IDFT of Y [k] as follows
   a. Compute the DFT of Y ∗[k]
   b. Take the complex conjugate
   c. Divide by 1/N
7. Save the first (L + P − 1) values of y[n].

Example:

Given:

x [n] = δ [n] + δ [n-1] – δ [n – 2] + δ [n – 3] + δ [n – 4] + δ [n – 5]

h[n] = – δ [n] + 2δ [n – 1] – δ [n – 2]

Find y[n] using Linear Convolution

Solution:

x[n] = {1, 1, –1, 1, 1, 1} Length L = 6

h[n] = {– 1, 2, –1} Length M =3

y[n] = x [n] ∗ h [n], Length N = L + M – 1 = 8

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]\, h[n-m]$$

(i) $n = 0,\ y[0] = \sum_{m=0}^{\infty} x[m]h[n-m]$

$y[0] = (1)(-1) = -1$

(ii) $n = 1,\ y[1] = \sum_{m=0}^{\infty} x[m]\, h[1-m]$

$y[1] = (1)(2) + (1)(-1) = 1$

(iii) $n = 2,\ y[2] = \sum_{m=0}^{\infty} x[m]\, h[2-m]$

$y[2] = (1)(-1) + (1)(2) + (-1)(-1) = 2$

(iv) $n = 3, y(3) = \sum_{m=0}^{\infty} x[m]\, h(3-m)$

$y[3] = (1)(-1) + (-1)(2) + (1)(-1) + 0 = -4$

(v) $n = 4, y(4) = \sum_{m=0}^{\infty} x(m) h(4-m)$

$y[4] = (-1)(-1) + (1)(2) + (1)(-1) = 3$

(vi) $n = 5\ y(5) = \sum_{m=0}^{\infty} x(m) h(5-m)$

$y[5] = (1)(-1) + (1)(2) + (1)(-1) + 0 = 0$

(vii) $n = 6, y(6) = \sum_{m=0}^{\infty} x(m)\, h(6-m)$

$y[6] = (1)(-1) + (1)(2) + 0 = 1$

(viii) $n = 7\ y(7) = \sum_{m=0}^{\infty} x(m)\ h(7-m)$

$y[7] = (1)(-1) + 0 = -1$

Therefore,

$$y[n] = \{ \underset{\uparrow}{-1}, 1, 2, -4, 3, 0, 1, -1 \}$$

Q3. Explain circular convolution using DFT

Ans: **[P | High]**

CIRCULAR CONVOLUTION:

1. The circular convolution, also known as **cyclic convolution**.
2. A convolution operation that contains a circular shift is called circular convolution.
3. Circular convolution of two sequences x1[n] and x2[n] is given by

$$y(m) = \sum_{n=0}^{N-1} x1(n)\, x2\,(m-n)_N$$

4. Circular convolution returns same number of elements that of two signals.

METHODS OF CIRCULAR CONVOLUTION:

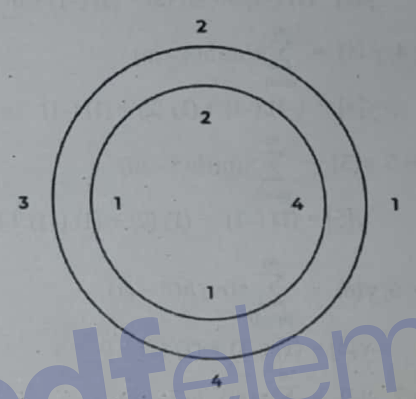
I) Concentric Circle Method:

1. Let x1(n) and x2(n) be two given sequences.

2. Steps followed for circular convolution of x1(n) and x2(n) are:
   a. Consider two concentric circles and plot N samples of x1(n) on the circumference of the outer circ (maintaining equal distance successive points) in anti-clockwise direction.
   b. To plot x2(n), plot N samples of x2(n) in clockwise direction on the inner circle, starting sampl placed at the same point as 0th sample of x1(n)
   c. Multiply equivalent samples on the two circles and add them to get output.
   d. Rotate the inner circle anti-clockwise with one sample at a time.
3. **Example:**

Consider $x_1(n) = \{1, 2, 3, 4\}$ & $x_2(n) = \{4, 1, 1, 2\}$

Now we take two concentric circles. Plot $x_1(n)$ on the outer circle in the counter clockwise direction an $x_2(n)$ on the inner circle in the clockwise direction.



We multiply the corresponding samples and add. This gives us y(0)

$\therefore y(0) \quad = 4 \times 1 + 2 \times 2 + 1 \times 3 + 1 \times 4$

$= 15$

We now rotate the inner circle in the anti-clock wise direction by one step. This gives us,

$\therefore y(1) \quad = 1 \times 1 + 4 \times 2 + 2 \times 3 + 1 \times 4$

$= 19$

We continue rotating the inner circle in steps of one.

$\therefore y(2) \quad = 1 \times 1 + 1 \times 2 + 4 \times 3 + 2 \times 4$

$= 23$

$\therefore y(3) \quad = 2 \times 1 + 1 \times 2 + 1 \times 3 + 4 \times 4$

$= 23$

If we now rotate the inner circle, we come back to the starting point. Hence we stop here.

$\therefore y(n) \quad = \{15, 19, 23, 23\}$

II) Matrix Multiplication Method:

1. Matrix method represents the two given sequence x1(n) and x2(n) in matrix form.
2. One of the given sequences will be repeated via circular shift of one sample at a time to form a N X N matrix.
3. Other sequence will be represented as column matrix.
4. Multiplication of two matrices gives the result of circular convolution
5. Example:

We generate a circular matrix of $x_2(n)$ and multiply by $x_1(n)$

$y(n) = x(n) \times x_2(n)$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 1 \\ 1 & 4 & 2 & 1 \\ 1 & 1 & 4 & 2 \\ 2 & 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

We get,

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 15 \\ 19 \\ 19 \\ 23 \end{bmatrix}$$

i.e. $y(n) = \{15, 19, 19, 23\}$

CHAP - 3: FAST FOURIER TRANSFORM

Q1. Write short notes on FFT & need of FFT

Ans:

[P | Medium]

FFT:

1. FFT stands for **Fast Fourier Transform**.
2. A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of sequence, or its inverse (IDFT).
3. Fourier analysis converts a signal from its original domain (often time or space) to a representation the frequency domain and vice versa.
4. The Fast Fourier Transform (FFT) is simply a fast (computationally efficient) way to calculate the Discrete Fourier Transform (DFT).

FFT WORKING:

1. By making use of periodicities in the sines that are multiplied to do the transforms, the FFT great reduces the amount of calculation required.
2. Functionally, the FFT decomposes the set of data to be transformed into a series of smaller data sets be transformed.
3. Then, it decomposes those smaller sets into even smaller sets.
4. At each stage of processing, the results of the previous stage are combined in special way.
5. Finally, it calculates the DFT of each small data set.
6. For example, an FFT of size 32 is broken into 2 FFTs of size 16, which are broken into 4 FFTs of size which are broken into 8 FFTs of size 4, which are broken into 16 FFTs of size 2.
7. Calculating a DFT of size 2 is trivial.

FFT EXAMPLE:

1. The main advantage of having FFT is that through it, we can design the FIR filters.
2. Mathematically, the FFT can be written as follows;

$$x[K] = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}$$

3. We have considered eight points named from x_0 to x_7.
4. We will choose the even terms in one group and the odd terms in the other.
5. Diagrammatic view of the above said has been shown in figure 3.1.

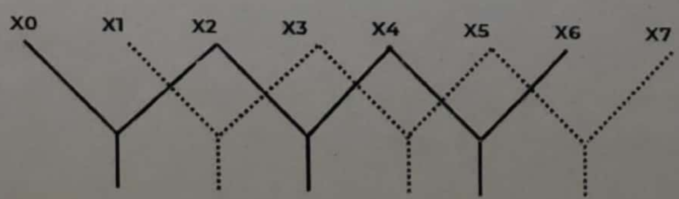


Figure 3.1

Here, points x0, x2, x4 and x6 have been grouped into one category and similarly, points x1, x3, x5 and x7 has been put into another category.

Now, we can further make them in a group of two and can proceed with the computation.

Now, let us see how these breaking into further two is helping in computation.

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r]W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]W_N^{(2r+1)k}$$

$$= \sum_{r=0}^{\frac{N}{2}-1} x[2r]W_{N/2}^{rk} + \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]W_{N/2}^{rk} \times W_N^k$$

$$= G[k] + H[k] \times W_N^k$$

Initially, we took an eight-point sequence, but later we broke that one into two parts G[k] and H[k].

G[k] stands for the even part whereas H[k] stands for the odd part.

If we want to realize it through a diagram, then it can be shown as below in figure 3.2

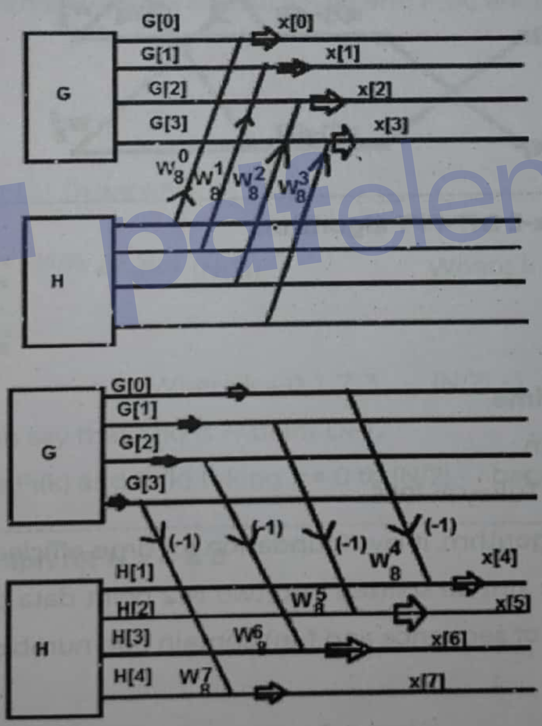


Figure 3.2

2. From the above figure, we can see that

$$W_8^4 = -1$$

$$W_8^5 = -W_8^1$$

$$W_8^6 = -W_8^2$$

$$W_8^7 = -W_8^3$$

13. Similarly, the final values can be written as follows

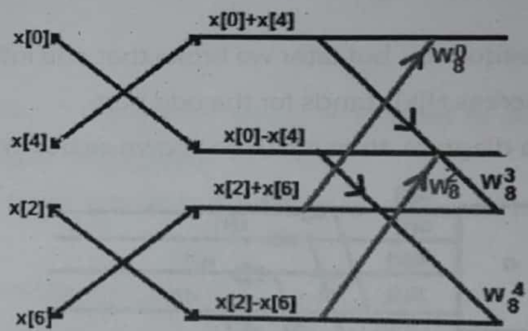
$$G[0] - H[0] = x[4]$$

$$G[1] - W_8^1 H[1] = x[5]$$

$$G[2] - W_8^2 H[2] = x[6]$$

$$G[1] - W_8^3 H[3] = x[7]$$

14. The above one is a periodic series.
15. The disadvantage of this system is that K cannot be broken beyond 4 point.
16. Now Let us break down the above into further.
17. We will get the structures something like this



Q2. Write short notes on Radix-2 DIT-FFT algorithm

Ans: [P | Medium]

RADIX 2 DIT FFT ALGORITHM:

1. DIT stands for **Decimation in Time**.
2. Radix-2 is the first FFT algorithm.
3. It was proposed by Cooley and Tukey in 1965.
4. Though it is not the efficient algorithm, it lays foundation for time-efficient DFT calculations.
5. Let the N point data sequence x(n) be splitted into two N/2 point data sequences $f_1(n)$ and $f_2(n)$ such that $f_1(n)$ contain even number of sequence and $f_2(n)$ contain odd number of sequence.
6. Therefore, we can write that

$$\left.\begin{aligned} f_1(n) &= x(2n) \qquad n = 0,\ 1,\ 2, \ldots\ldots\ldots, \frac{N}{2}-1 \\ f_2(n) &= x(2n+1) \qquad n = 0,\ 1,\ 2, \ldots\ldots\ldots, \frac{N}{2}-1 \end{aligned}\right]$$

7. Above equations shows that the time domain sequence is splitted into two sequences.
8. The above splitting operation is called decimation.
9. It is called decimation in time domain because it is done on time domain sequence.
10. The N point DFT of x(n) is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad Where\ k = 0, 1, 2, 3 \ldots\ldots N-1$$

Remove Watermark Now

. Splitting the sequence x(n) into even and odd number of sequence, we get

$$X(k) = \sum_{n\ even} x(n)W_N^{kn} + \sum_{n\ odd} x(n)\ W_N^{kn}$$

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} x\ (2m)\ W_N^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x\ (2m+1)\ W_N^{(2m+1)k}$$

. In above equation, let put $f_1(m) = x(2m)$ and $f_2(m) = x(2m + 1)$.

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} f_1(m)\ (W_N^{mk})^2 + \sum_{m=0}^{\frac{N}{2}-1} f_2(m)\ (W_N^{mk})^2\ W_N^k$$

. Since $W_N^2 = W_{\frac{N}{2}}$ we have

$$X(k) = \sum_{m=0}^{\frac{N}{2}-1} f_1(m)\ W_{\frac{N}{2}}^{mk} + W_N^k + \sum_{m=0}^{\frac{N}{2}-1} f_2(m)\ W_{\frac{N}{2}}^{mk}$$

. $X(k) = F_1(k) + W_N^k + F_2(k)$ Where k = 0, 1, 2, 3, ... N – 1 .... Eq (1)

. Where $F_1(k)$ is the N/2 point of DFT $F_1(m)$ and $F_2(k)$ is the N/2 point of DFT $F_2(m)$

$F_1(k)$ and $F_2(k)$ are periodic with period N/2 because $F_1(k)$ and $F_2(k)$ are N/2 point DFTs.

Hence we can write,

$F_1(k + (N/2)) = F_1(k)$

$F_2(k + (N/2)) = F_2(k)$

Substituting k by k + (N/2) in Eq. (1) we get,

$X(k + (N/2)) = F_1(k + (N/2)) + W_N^{k+(N/2)} + F_2(k + (N/2))$ Where k = 0, 1, 2, 3, ... N – 1

. We know that $W_N^{k+\left(\frac{N}{2}\right)} = -W_N^k$

$X(k + (N/2)) = F_1(k) - W_N^k\ F_2(k)$ Where k = 0, 1, 2, 3, ..., (N/2) - 1

. From above equation, we can say that X(k) is *N*-point DFT.

. Therefore, we can determine $F_1(k)$ and $F_2(k)$ taking $k = 0$ to (N/2) - 1 because they are N/2 point DFTs.

---

3. Explain DIT FFT Flow Graph for N = 4 & 8

ns: **[P | Medium]**

RST STAGE OF DECIMATION:

Figure 3.3 shows that 8-point DFT can be determined directly.

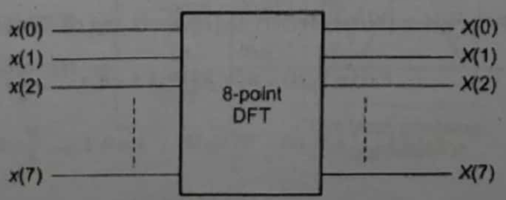


Figure 3.3

X(k) can be obtained From $F_1(k)$ and $F_2(k)$ where $F_1(k)$ and $F_2(k)$ are two 4-point DFTs.

The symbolic diagram operation has been shown in figure 3.4.

The sequences of $F_1(m)$ and $F_2(m)$ are given below:

$$\left.\begin{aligned} f_1(m) &= x(2n) = \{x(0), x(2), x(4), x(6)\} \\ f_2(m) &= x(2n+1) = \{x(1), x(3), x(5), x(7)\} \end{aligned}\right\} - \frac{N}{2} \text{ point sequences}$$

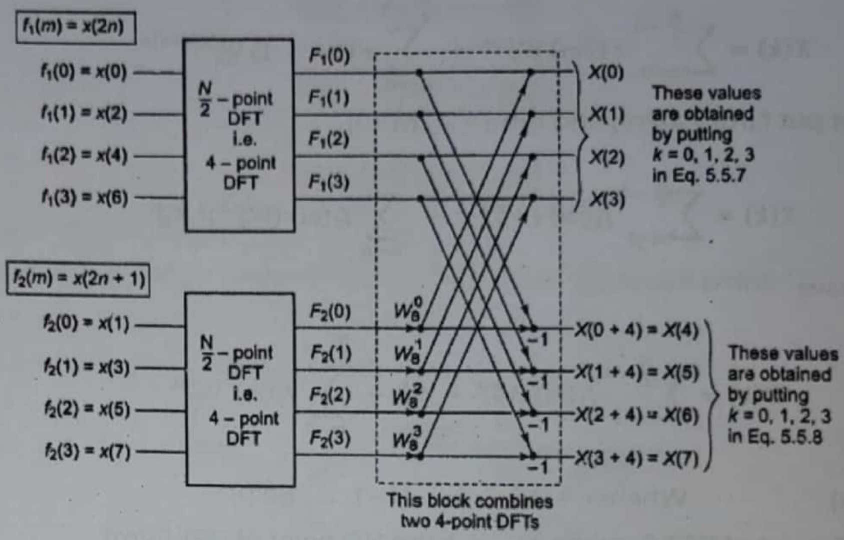


Figure 3.4

SECOND STAGE OF DECIMATION:

1. Let us split $f_1(n)$ into the following even and odd numbered samples:

$$\left.\begin{aligned} v_{11}(n) &= f_1(2n) & n &= 0, 1, \ldots\ldots, \frac{N}{4} \\ v_{12}(n) &= f_1(2n+1) & n &= 0, 1, \ldots\ldots, \frac{N}{4} \end{aligned}\right\}$$

2. $v_{11}(n)$ and $v_{12}(n)$ are the even and odd numbered sequences of $f_1(n)$ containing (N/4) samples because $f_1(n)$ contains (N/2) samples.
3. Now let us split $f_2(n)$ into the following even and odd numbered samples:

$$\left.\begin{aligned} v_{21}(n) &= f_1(2n) & n &= 0, 1, \ldots\ldots, \frac{N}{4} \\ v_{22}(n) &= f_1(2n+1) & n &= 0, 1, \ldots\ldots, \frac{N}{4} \end{aligned}\right\}$$

4. $v_{21}(n)$ and $v_{22}(n)$ are the even and odd numbered sequences of $f_2(n)$ containing (N/4) samples because $f_2(n)$ contains (N/2) samples.
5. Earlier we have obtained $X(k)$ and X(k + (N/2)) from $F_1(k)$ and $F_2(k)$ with $f_1(n)$ and $f_2(n)$ as decimated sequences and the length of DFT was (N/2).
6. As before, we can obtain $F_1(k)$ and F_1(k + (N/4)) from $v_{11}(n)$ and $v_{12}(n)$. Therefore, we can write that

$$F_1(k) = V_{11}(k) + W_{\frac{N}{2}}^{k} V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1$$

and

$$F_1\left(k + \frac{N}{4}\right) = V_{11}(k) - W_{\frac{N}{2}}^{k} V_{12}(k) \qquad k = 0, 1, \ldots, \frac{N}{4} - 1$$

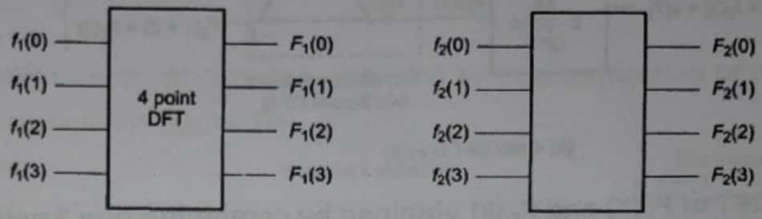
7. In above equations, the (N/2) point DFT are obtained from (N/4) point DFT.
8. $V_{11}(k)$ and $V_{12}(k)$ are the (N/4) point DFT of $v_{11}(n)$ and $v_{12}(n)$ respectively.
9. Similarly, we can write the following for $F_2(k)$ also:

$$F_2(k) = V_{21}(k) + W_{\frac{N}{2}}^{k} V_{22}(k) \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

and

$$F_2\left(k + \frac{N}{4}\right) = V_{21}(k) - W_{\frac{N}{2}}^{k} V_{22}(k) \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

0. In above equations, the (N/2) point DFT are obtained from (N/4) point DFT.
1. $V_{21}(k)$ and $V_{22}(k)$ are the (N/4) point DFT of $v_{21}(n)$ and $v_{22}(n)$ respectively.
2. Figure 3.5 shows $F_1(k)$ and $F_2(k)$ are two 4 points DFT which are shown by two separate blocks.



(a) $F_1(x)$ (b) $F_2(x)$

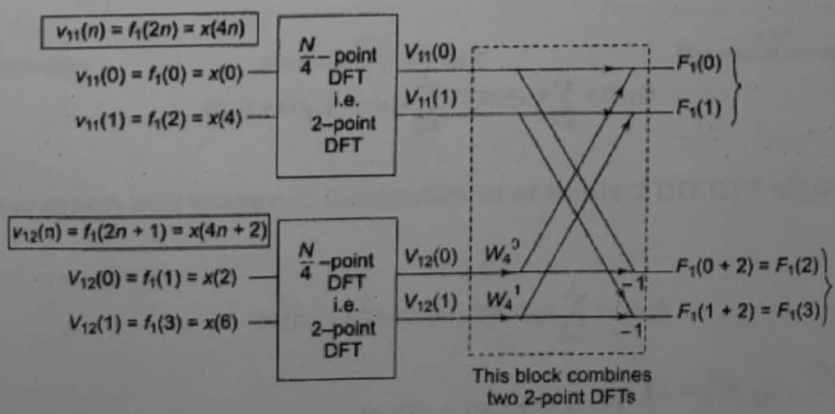
Figure 3.5: A Symbolic Representation of 4 point DFT using direct computation

3. For $N = 8$, $V_{11}(k)$ and $V_{12}(k)$ are (N/4) = 2 point DFTs.
4. The symbolic diagram of operation has been shown in Figure 3.6
5. From Figure 3.6(a) the sequences $v_{11}(n)$ and $v_{12}(n)$ for 4 point $f_1(n)$ are given by,

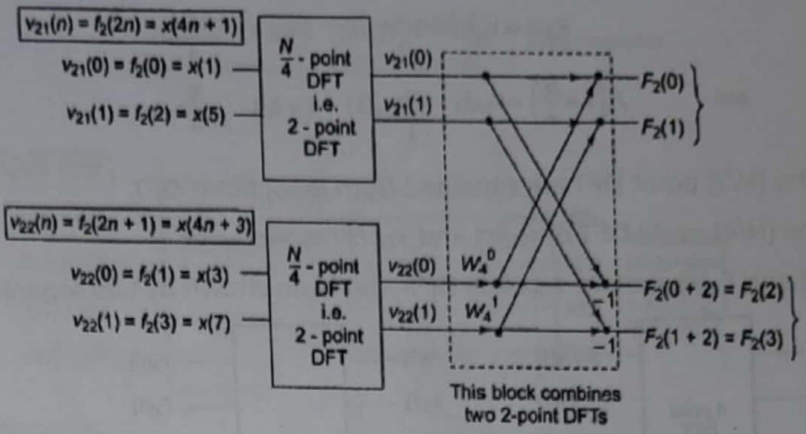
$$\left.\begin{aligned} v_{11}(n) &= f_1(2n) = x(4n) = \{x(0), x(4)\}, n = 0, 1 \\ v_{12}(n) &= f_1(2n+1) = x(4n+2) = \{x(2), x(6)\}, n = 0, 1 \end{aligned}\right\} - \frac{N}{4} \text{ point sequences}$$

6. Therefore, the two (N/4) point DFT can be computed separately and combining them we will get (N/2) point DFT.
7. The symbolic diagram of an operation has been shown in Figure 3.6(b).
8. From Figure 3.6(b) the sequences $v_{21}(n)$ and $v_{22}(n)$ for 4 point $f_2(n)$ are given by,

$$\left.\begin{aligned} v_{21}(n) &= f_1(2n) = x(4n+1) = \{x(1), x(5)\}, n = 0, 1 \\ v_{12}(n) &= f_1(2n+1) = x(4n+3) = \{x(3), x(7)\}, n = 0, 1 \end{aligned}\right\} - \frac{N}{4} \text{ point sequences}$$



(a) 4 pint DFT of $F_1(k)$



(b) 4 pint DFT of $F_1(k)$

Figure 3.6: Point DFT of F_1 (X) and F_2 (k) obtained by combining two 2 point DFTs.

THIRD STAGE OF DECIMATION:

1. Let us further split (N/4) point sequences in their even and odd parts.
2. Therefore, we will get the next stage of decimation and the sequences which will be of length (N/8) point. (N/8) = 1 for $N = 8$.
3. Figure 3.7 shows the two-point DFTs.

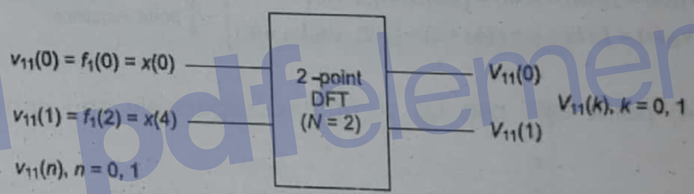


Figure 3.7: Computation of 2-point DFT v11 (k)

4. Let us compute $V_{11}(k)$ of figure 3.6(a). From figure 3.7 get

$$V_{11}(k) = \sum_{n=0}^{1} v_{11}(n) W_2^{kn},\ k = 0, 1$$

5. For $k = 0$,

$$V_{11}(0) = \sum_{n=0}^{1} v_{11}(n) W_2^0 = \sum_{n=0}^{1} v_{11}(n) = v_{11}(0) + v_{11}(1)$$

6. Because $W_2^0 = 1$

For $k = 1$,

$$V_{11}(1) = \sum_{n=0}^{1} v_{11}(n) W_2^n = v_{11}(0) W_2^0 + v_{11}(1) W_2^1$$

We know that W_2^0 and $W_2^1 = -1$. Now we can write as

$$V_{11}(1) = V_{11}(0) - V_{11}(1)$$

Since $W_2^0 = 1$, let us rewrite above equations in the following forms:

$$\left.\begin{aligned} V_{11}(0) &= v_{11}(0) + W_2^0\ v_{11}(1) \\ V_{11}(1) &= v_{11}(0) - W_2^0\ v_{11}(1) \end{aligned}\right\}$$

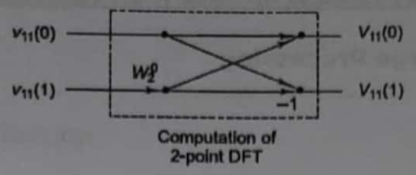


Figure 3.8

- Figure 3.8 shows signal flow graph for third stage. It cannot be decimated further because it is a two-point DFT.
- The complete signal flow graph which is obtained by interconnection of these individual stage wise signal flow graph is shown in figure 3.9

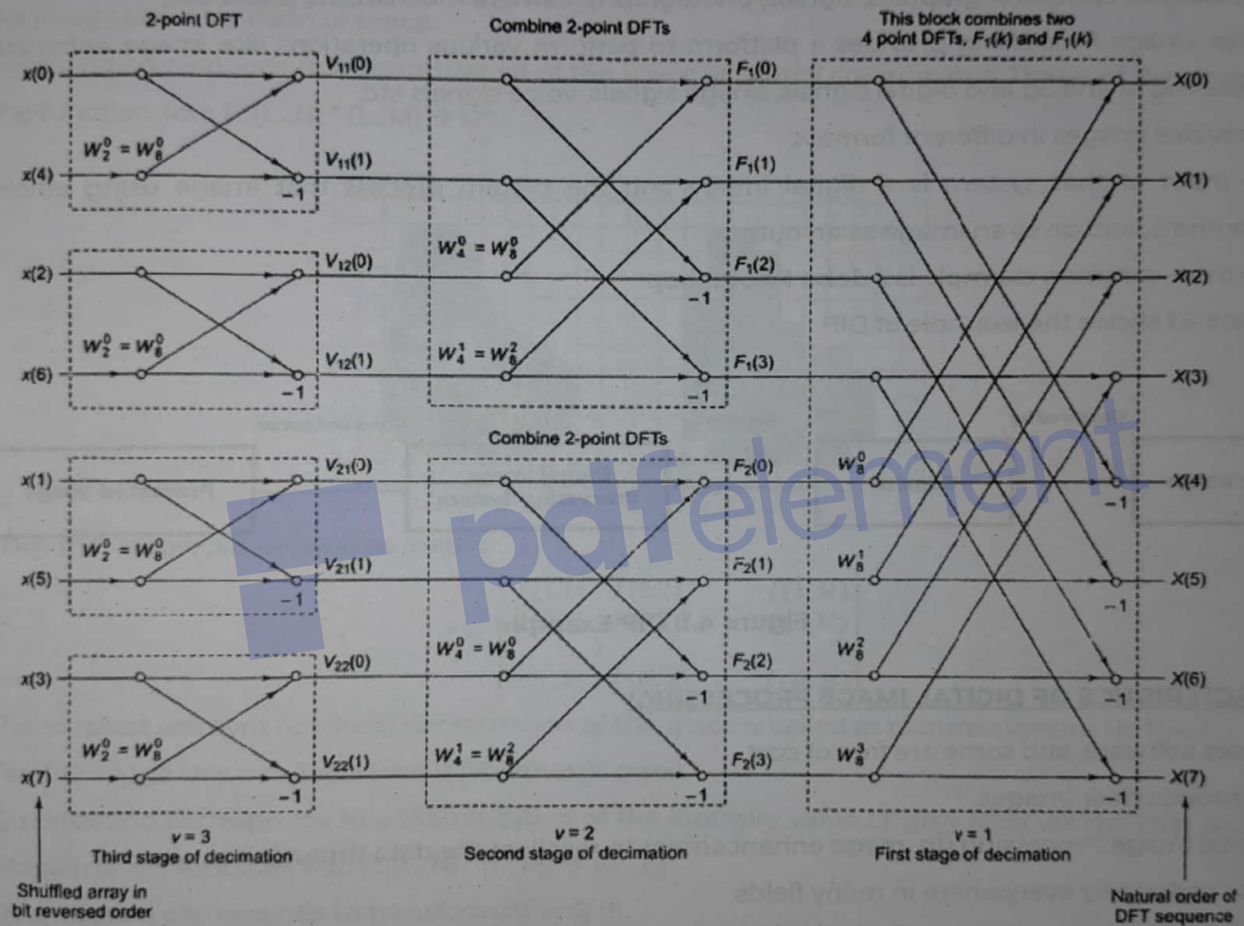


Figure 3.9: Signal flow graph and stages of Computation of Radix-2 DIT DFT algorithm for N = 8.

CHAP - 4: DIGITAL IMAGE FUNDAMENTALS

Q1. Write short notes on Digital Image Processing

Ans:

[P | Hig

DIGITAL IMAGE PROCESSING:

1. Digital image processing (DIP) deals with manipulation of digital images through a digital compute
2. It is a subfield of signals and systems.
3. It focus particularly on images.
4. Digital Image Processing is a software which is used in image processing.
5. For example: computer graphics, signals, photography, camera mechanism, pixels, etc.
6. Digital Image Processing provides a platform to perform various operations like image enhancin processing of analog and digital signals, image signals, voice signals etc.
7. It provides images in different formats.
8. The input of that system is a digital image and the system process that image using efficie algorithms, and gives an image as an output.
9. The most common example is **Adobe Photoshop**.
10. Figure 4.1 shows the example of DIP.

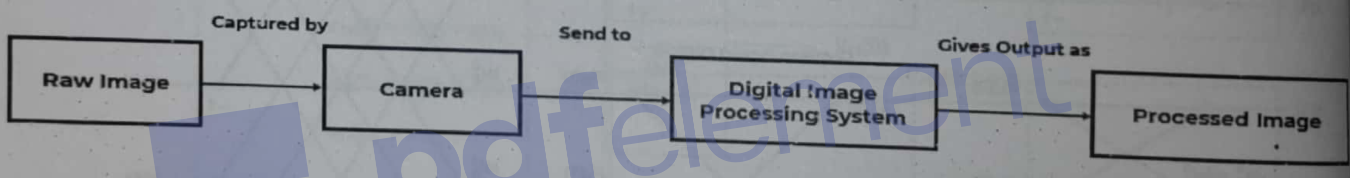


Figure 4.1: DIP Example

CHARACTERISTICS OF DIGITAL IMAGE PROCESSING:

1. It uses software, and some are free of cost.
2. It provides clear images.
3. Digital Image Processing do image enhancement to recollect the data through images.
4. It is used widely everywhere in many fields.
5. It reduces the complexity of digital image processing.
6. It is used to support a better experience of life.

ADVANTAGES:

1. Image reconstruction (CT, MRI, SPECT, PET)
2. Image reformatting (Multi-plane, multi-view reconstructions)
3. Fast image storage and retrieval
4. Fast and high-quality image distribution.
5. Controlled viewing (windowing, zooming)

DISADVANTAGES:

1. It is very much time-consuming.

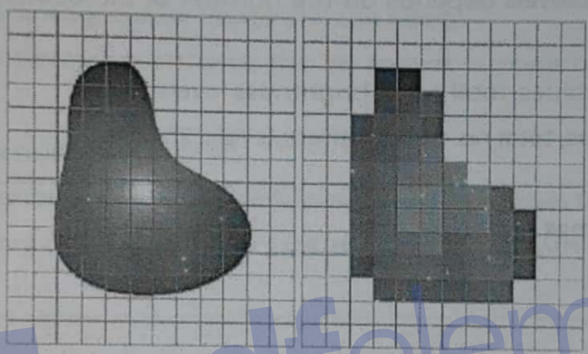
2. It is very much costly depending on the particular system.
3. Qualified persons can be used.

Q2. Explain Sampling & Quantization

Ans: **[P | High]**

SAMPLING & QUANTIZATION:

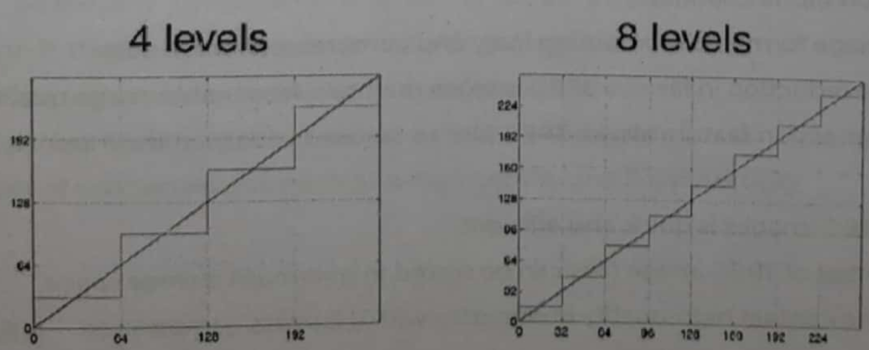
1. Sampling and Quantization are two important processes use to convert continuous analog signal into digital image.
2. Sampling and Quantization deals with integer values.
3. An image is the function of space.
4. Sampling corresponds to a discretization of the space or spatial co-ordinates. That is of the domain of the function, into F: [1....N] * [1....M] → R^m



5. Thus the image can be seen as matrix.

$$f = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,M) \\ f(2,1) & f(2,2) & & f(2,M) \\ \vdots & & \ddots & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,M) \end{bmatrix}$$

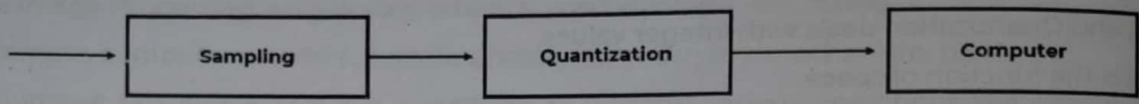
6. The smallest element resulting discretization of the space is called as picture element i.e. Pixel
7. For 3-D image this is called as voxel (Volumetric pixel).
8. Quantization corresponds to a discretization of the intensity value or grey level values. That is of the domain of the function, into F: [1....N] * [1....M] → [0....L]
9. Quantization corresponds to transformation Q (f)



10. Typically, 256 levels (8 bits/pixel) suffice to represent the intensity.
11. For color images, 256 levels are usually used for each of the color intensity.

CONVERTING AN ANALOG IMAGE INTO DIGITAL IMAGE:

1. Let F (X, Y) denotes input picture that we want to convert to digital form.
2. Here F represents the brightness at spatial co-ordinates x and y.
3. An input image may be continuous with respect to the x & y Co-ordinates, and also in amplitude.
4. To convert it to digital form, we have to sample the function in both co-ordinates and in amplitude
5. Digitization the co-ordinates value is called SAMPLING.
6. Digitization the amplitude value is called QUANTIZATION.



7. Every sampled pixel values are quantized to the nearest brightness levels.
8. Total number of brightness levels depends on the number of bit locations used for storage of p value.
9. For example, for 2 bit pixel values, total no. of brightness levels. $2^3 = 8$.
10. Similarly, for 8 bit pixel values, total no. of brightness levels, 2^8- 256

Q3. Explain Image File Format

Ans: **[P | Medi**

IMAGE FILE FORMAT:

1. Image Format describes how data related to the image will be stored.
2. Data can be stored in compressed, Uncompressed or vector format.
3. Each format of the image have a different advantage and disadvantage.
4. Image types such a TIFF are good for printing while JPG or PNG, are best for web.

JPEG:

1. JPEG stands for **Joint Photographic Experts Group.**
2. It is used to display photographs and other continuous tone images in HTML documents over WW
3. It supports CMYK, RGB & Greyscale color modes.
4. It does not support alpha channel.
5. It's a standard image format for containing lossy and compressed image data.
6. Despite the huge reduction in file size JPEG images maintain reasonable image quality.
7. This unique compression feature allows JPEG files to be used widely on the Internet, Computers, a Mobile Devices.
8. The sharing of JPEG images is quick and efficient.
9. Also, a large number of JPEG image files can be stored in minimum storage space.
10. JPEG files can also contain high-quality image data with a lossless compression.

BMP:

1. The BMP extension represents **Bitmap Image file.**

2. It is a standard windows image format on DOS and windows – compatible computers.
3. It supports RGB, Indexed Color, Greyscale and Bitmap color modes.
4. It does not support alpha channel.
5. BMP file contains raster graphics data which are independent of display devices.
6. That means a BMP image file can be viewed without a graphics adapter.
7. BMP images are generally uncompressed or compressed with a lossless compression method.
8. The files can store two-dimensional digital images with both monochrome and color.

TIFF:

1. TIFF stands for **Tagged Image File Format**.
2. It is used to exchange files between applications and computer platforms.
3. It is a standard file format that is largely used in the publishing and printing industry.
4. The extensible feature of this format allows storage of multiple bitmap images having different pixel depths, which makes it advantageous for image storage needs.
5. Since it introduces no compression artifacts, the file format is preferred over others for archiving intermediate files.
6. A TIFF file uses the file extension ".tif" or ".tiff".
7. TIFF is also known for its flexibility and has more features and capabilities than other image file formats.
8. Most image editing software or applications are capable of working with TIFF files.

Q4. Explain Connectivity **[P | Medium]**

Ans:

CONNECTIVITY:

1. In binary valued digital imaging, a pixel can either have a value of 1 (when it's part of the pattern), or 0 (when it's part of the background) i.e. there is no grayscale level.
2. We will assume that pixels with value 1 are black while zero valued pixels are white.
3. In order to identify **objects** in a digital pattern, we need to locate groups of black pixels that are "connected" to each other.
4. In other words, the **objects** in a given digital pattern are the **connected components** of that pattern.
5. In general, a connected component is a set of black pixels, P, such that for every pair of pixels p_i and p_j in P, there exists a sequence of pixels $p_i, ..., p_j$ such that:
   a. All pixels in the sequence are in the set P i.e. are black, and
   b. Every 2 pixels that are adjacent in the sequence are "neighbors"
6. There are 2 types of connectedness, namely: 4-connectivity and 8-connectivity.

TYPES:

I) 4 – Connectivity:

1. A pixel, Q, is a 4-neighbor of a given pixel, P, if Q and P share an edge.
2. The 4-neighbors of pixel P (namely pixels P2, P4, P6 and P8) are shown in Figure 4.2.

3. A set of black pixels, P, is a 4-connected component if for every pair of pixels p_i and p_j in P, there exis a sequence of pixels pi, ..., pj such that:
   a. All pixels in the sequence are in the set P i.e. are black, and
   b. Every 2 pixels that are adjacent in the sequence are 4-neighbors.

Figure 4.2

Examples of 4-connected patterns:

The following figure 4.3 are examples of patterns that are 4-connected.

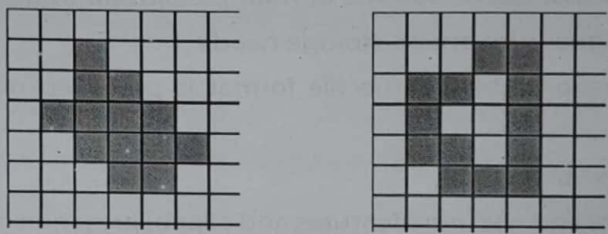


Figure 4.3

II) 8 – Connectivity:

1. A pixel, Q, is an 8-neighbor (or simply a neighbor) of a given pixel, P, if Q and P either share an edge a vertex.
2. The 8-neighbors of a given pixel P make up the Moore neighborhood of that pixel.
3. A set of black pixels, P, is an 8-connected component (or simply a connected component) if for eve pair of pixels p_i and p_j in P, there exists a sequence of pixels p_i, ..., p_j such that:
   a. All pixels in the sequence are in the set P i.e. are black, and
   b. Every 2 pixels that are adjacent in the sequence are 8-neighbors
4. All 4-connected patterns are 8-connected i.e. 4-connected patterns are a subset of the set of connected patterns.
5. On the other hand, an 8-connected pattern may not be 4-connected.

Example of 8-connected pattern:

The figure 4.4 below is an example of a pattern that is 8-connected but not 4-connected:

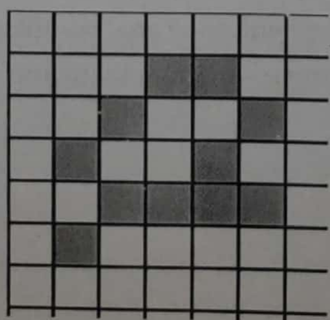


Figure 4.4

Example of a pattern that's not 8-connected:

The figure 4.5 below is an example of a pattern that is not 8-connected i.e. is made up of more than one connected component (there are 3 connected components in the figure below)

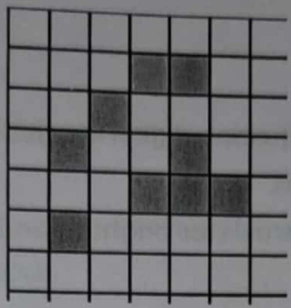


Figure 4.5

CHAP - 5: IMAGE ENHANCEMENT IN SPATIAL DOMAIN

Q1. Explain image enhancement in spatial domain

Ans: **[P | High]**

IMAGE ENHANCEMENT:

1. Image enhancement is the process of adjusting digital images so that the results are more suitable fo[r] display of image or further image analysis.
2. For example, you can remove noise, sharpen, or brighten an image, making it easier to identify ke[y] features.
3. Image enhancement includes gray level and contrast manipulation, noise reduction, edge sharpening filtering, interpolation etc.
4. The enhancement process does not increase the inherent information content in the data.
5. But it does increase the dynamic range of the chosen features so that they can be detected easily.

SPATIAL DOMAIN INCLUDES:

I) Point processing:

1. In point processing, single processor are used. I.e. 1*1 operators.
2. It is also called as Zero memory point operator.
3. In this output image pixel value is obtained directly from processing the input image pixel value.
4. For every input pixel value, transformation function gives corresponding output image pixel value and no memory location is required to store intermediate results.
5. It includes:
   a. Digital Negative
   b. Contrast stretching
   c. Thresholding
   d. Grey Level Slicing
   e. Bit Plane Slicing
   f. Dynamic Range Compression
   g. Power Law Transformation

II) Neighborhood processing:

1. It is also called as **Spatial Domain technique**.
2. Spatial Filtering involves passing a weighted mask or the kernel over the image.
3. Then replacing the original pixel value corresponding to the center of the kernel with the sum of the original pixel values in the region corresponding to the kernel multiplies by the kernel weight.
   a. **Smoothing Linear Filters:** Example of Linear filters is Low Pass Averaging Filter, Weighted Average Filter and Trimmed Average Filter.
   b. **Smoothing Non-Linear Filters:** Non- Linear Filters are also called as Ordered Statistic Filters Example is Median, Max and Min Filters.
   c. **First Order Derivative Filters:** Example is Robert, Prewit, Sobel and Fri-Chen Filter.
   d. **Sharpening Second Order Derivative Filters:** Example is Laplacian Filter, High Pass Filter and High Boost Filter.

II) Histogram Processing:

- It involves the modification of input image histogram so as to improve the visual quality of image on display device.
- It includes:

a. **Histogram Equalization:**
   - It is a process that attempts to spread out the gray levels in an image so that they are evenly distributed over entire range.
   - The histogram of the output image is almost uniform over the entire range of gray levels.
   - It provides only one type of output.

b. **Histogram Specification:**
   - It is a process that attempts to spread out the gray levels in an image as per the specified image histogram.
   - Modifies histogram of the input image closely matches with the histogram of the specified image.

c. **Histogram Stretching:**
   - It is the process that attempts to spread out the gray levels in an image linearly as per the required range of output image histogram.

Q2. Explain grey level transformation

Ans: **[P | High]**

GREY LEVEL TRANSFORMATION:

- Gray level transformation is a significant part of image enhancement techniques which deal with images composed of pixels.
- All Image Processing Techniques focused on gray level transformation as it operates directly on pixels.
- The gray level image involves 256 levels of gray and in a histogram, horizontal axis spans from 0 to 255, and the vertical axis depends on the number of pixels in the image.
- The simplest formula for image enhancement technique is: s = T * r
- Where T is transformation, r is the value of pixels, s is pixel value before and after processing.
- The overall graph is shown below in figure 5.1

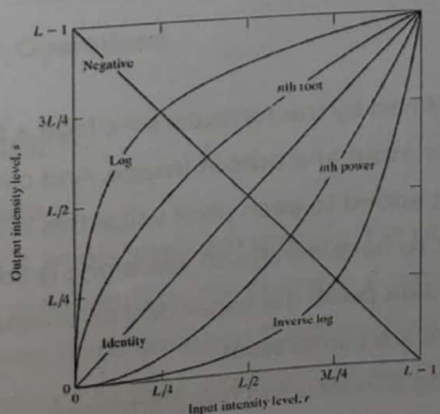


Figure 5.1: Overall Graph

I) Linear Transformation:

1. The linear transformation includes identity transformation and negative transformation.
2. **In identity transformation**, each value of the image is directly mapped to each other values of the output image.
3. Figure 5.2 shows the example of Identity transformation.

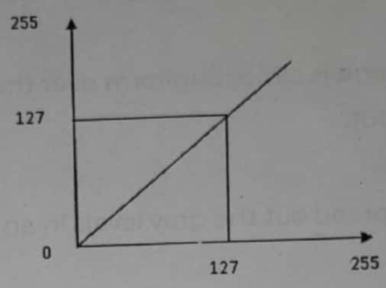


Figure 5.2: Identity Transformation

4. **Negative transformation** is the opposite of identity transformation.
5. Here, each value of the input image is subtracted from L-1 and then it is mapped onto the output image.
6. Figure 5.3 shows the example of negative transformation.

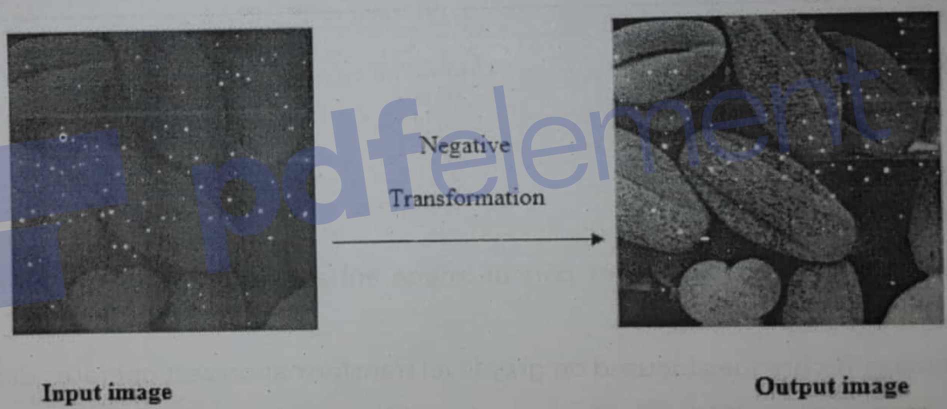


Figure 5.3: Negative Transformation

II) Logarithmic transformations:

1. Logarithmic transformation is divided into two types:
   a. Log transformation
   b. Inverse log transformation
2. The log transformations can be defined by this formula: $s = c \log(r + 1)$
3. Here, s and r are the pixel values for input and output image. And c is constant.
4. In the formula, we can see that 1 is added to each pixel value this is because if pixel intensity is zero in the image then log(0) is infinity so, to have minimum value one is added.
5. When log transformation is done dark pixels are expanded as compared to higher pixel values.
6. In log transformation higher pixels are compresses.
7. Figure 5.4 shows the example of log & inverse log transformation.

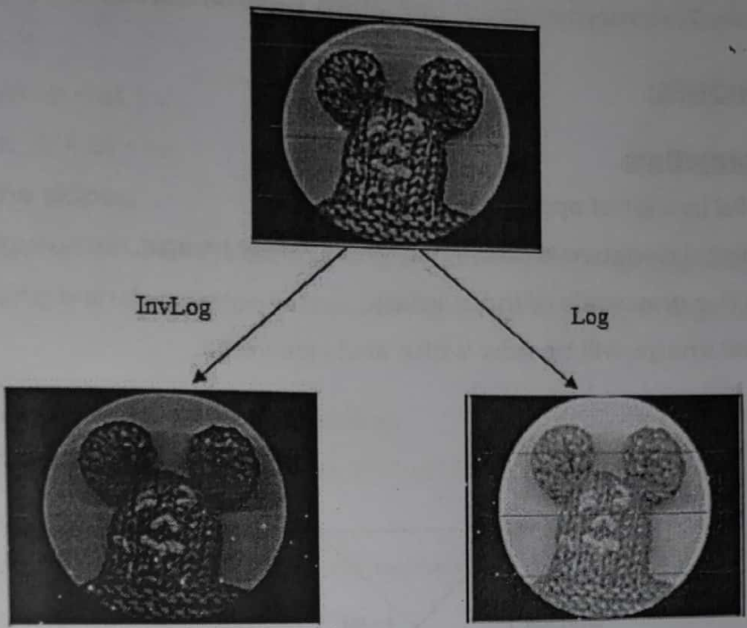
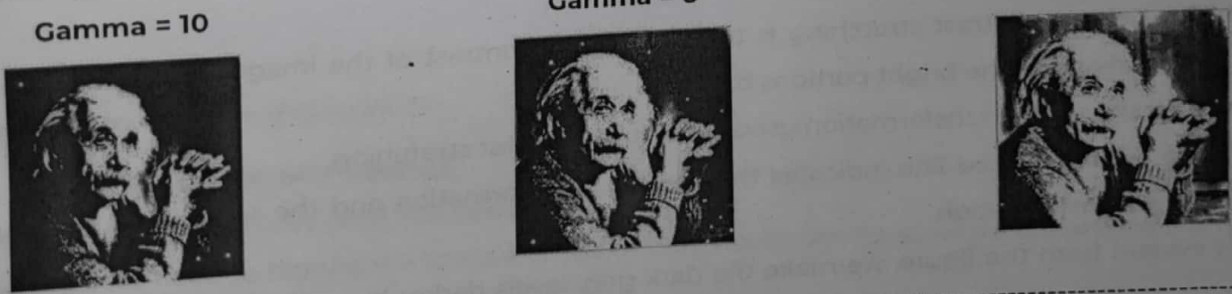


Figure 5.4: log & inverse log transformation

III) **Power - Law transformations:**

1. Power Law Transformation is of two types of transformation nth power transformation and nth root transformation.
2. **Formula:** s = cr ^ γ
3. Here, γ is gamma, by which this transformation is known as gamma transformation.
4. All display devices have their own gamma correction.
5. That is why images are displayed at different intensity.
6. These transformations are used for enhancing images.
7. **For example:**



Q3. **Explain Zero Memory Point Operations** **[P | High]**

Ans:

ZERO MEMORY POINT OPERATIONS:

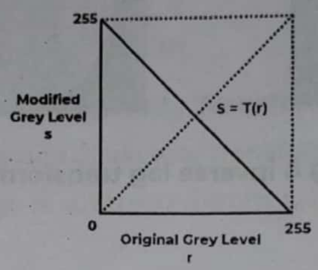
1. It is also known as **point processing**.
2. In Zero Memory Point Operations, single pixels are used. i.e. T is 1 x 1 operator.
3. It means that the new value f(x, y) depends on the operator T and the present f(x, y)
4. For every input image pixel value, Transformation function gives corresponding output image pixel value, no memory location is required to store intermediate results.
5. Let r denotes input image pixel value and S denotes output image pixel value.

6. Then S = T(r); where T is any Zero memory point operation Transformation function.

POINT PROCESSING TECHNIQUES:

I) Digital Negative Transformation:

1. Digital negatives are useful in a lot of applications.
2. A common example of digital negative is displaying of an X-Ray Image.
3. Digital Negative reverses the gray scale of input image.
4. That is black in the original image will be now white and vice versa.
5. It is defined as S = T(r)
6. Where T is the Digital Negative Tx Function



7. The digital negative image can be obtained by using simple transformation given by

$$s = 255 - r$$

8. Hence when r = 0 then s = 255 & when s = 0 then r = 255
9. In general, s = (L – 1) – r
10. Here L is the number of grey levels. (256 in this case)

II) Contrast Stretching:

1. Many times we obtain low contrast images due to poor illuminates or due to wrong setting of the lens aperture.
2. The idea behind contrast stretching is to increase the contrast of the images by making the dark portions darker and the bright portions brighter.
3. Figure 5.5 shows the transformation used to achieve contrast stretching.
4. In figure 5.5, the dotted line indicates the identity transformation and the solid line is the contrast stretching transformation.
5. As is evident from the figure, we make the dark grey levels darker by assigning a slope greater than one.
6. One can assign different slopes depending on the input image and the application.

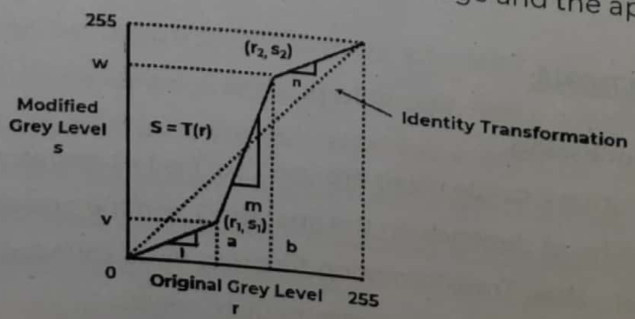


Figure 5.5

The formulation of the contrast stretching algorithm is given below:

s	$= l \cdot r$	$0 <= r < a$
	$= m \cdot (r - a) + v$	$a <= r < b$
	$= n \cdot (r - b) + w$	$b <= r < L - 1$

Where l, m & n are the slopes.

It is clear from the figure that l and n are less than one while m is greater than one.

The contrast stretching transformation increases the dynamic range of the modified image.

Thresholding:

Extreme contrast stretching yields **Thresholding**.

As shown in figure 5.5, in contrast stretching, first and last slope are made zero and the center slope is increased.

We would get threshold transformation and threshold function i.e. $r_1 = r_2$, $s_1 = 0$ and $s_2 = L - 1$

Figure 5.6 shows the example of Thresholding.

The formula for achieving Thresholding is as follows:

s	$= 0$	if $r <= a$
	$= L - 1$	if $r > a$

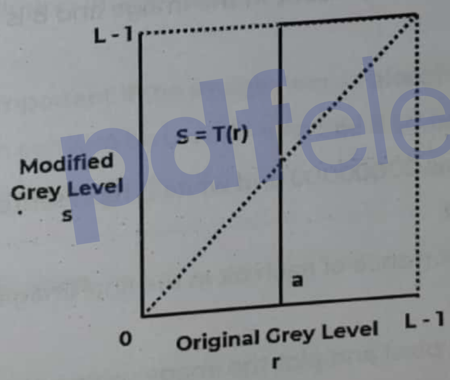


Figure 5.6

Where L is the number of grey levels.

Grey Level Slicing (Intensity Slicing):

Thresholding splits the grey level into two parts.

Sometimes we need to highlight a specific range of grey values like for example enhancing the flaws in an X-Ray or a CT Image.

In such circumstances, we use a transformation known as grey level slicing.

The transformation is shown in figure 5.7

This transformation looks similar to the Thresholding function except that here we select a band of grey level values.

6. This can be implemented using the formulation

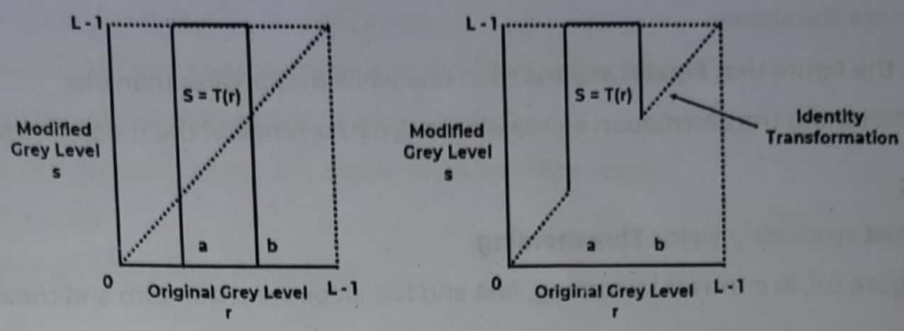
s	$= L - 1$	if $a <= r <= b$
	$= 0$	otherwise

7. This method is known as **grey level slicing without background**.
8. In some applications, we not only need to enhance a band of grey levels but also need to retain the background.

This technique of retaining the background is called as **grey level slicing with background.**

). This can be implemented using the formulation

$$s = L-1 \quad \text{if } a <= r <= b$$
$$= r \quad \text{otherwise}$$



(a) Slicing without background (b) Slicing with background

Figure 5.7

V) Bit plane slicing:

1. In this technique, we find out the contribution made by each bit to the final image.
2. Image is defined as a 256 x 256 x 8 image.
3. In this, 256 x 256 is the number of pixels present in the image and 8 is the number of bits required to represent each pixel.
4. 8 bits simply means 28 or 256 grey levels.
5. Now each pixels will be represented by 8 bits.
6. For example, black is represented as 00000000 and white is represented as 11111111 and between them, 254 grey levels are accommodated.
7. In bit plane slicing, we see the importance of each bit in the final image.
8. This can be done as follows:
   a. Consider the LSB value of each pixel and plot the image using only the LSBs.
   b. Continue doing this for each bit till we come to the MSB.
   c. Note that we will get 8 different images and all the 8 images will be binary.

VI) Dynamic range compression (Log Transformation):

1. Most of the times, the dynamic range of the image exceeds the capability of the display device.
2. What happens is that some pixel values are so large that the other low value pixels get obscured.
3. A simple day to day example of such a phenomena is that during daytime, we cannot see the stars.
4. The reason behind this is that the intensity of sun is so large and that of the stars is so low that the eye cannot adjust to such a large dynamic range.
5. In image processing, a classic example of such large differences in grey levels is the fourier spectrum.
6. In fourier spectrum only some of the values are very large while most of the values are too small.
7. The dynamic range of pixels is of the order of 10^6.
8. Hence, when we plot the fourier spectrum, we see only small dots, which represent the large values.
9. Sometimes needs to be done to able to see the small values as well.
10. This technique of compressing the dynamic range is known as dynamic range compression.
11. Dynamic range compression is achieved by using a log operator.
12. Figure 5.8 represent dynamic range compression.

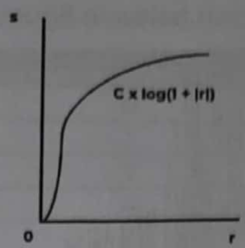


Figure 5.8

3. Where C is the normalization constant.

VII) Power Law Transformation:

1. The basic formula for power law transformation is

$$g(x, y) = c \times f(x, y)^{\gamma}$$

2. It can also be written as $s = cr^{\gamma}$
3. Here c and γ are positive constants.
4. The transformation is shown below for different values of γ which is also called the gamma correction factor.
5. By changing the value of gamma, we obtain a family of transformation curves.
6. Non linearities encountered during image capturing, printing and displaying can be corrected using gamma correction.
7. Hence gamma correction is important if the image needs to be displayed on the computer.
8. The power law transformation can also be used to improve the dynamic range of an image.
9. Refer Q2 (Power Law Transformation Part)

Q4. Explain Histogram Equalization **[P | Medium]**

Ans:

HISTOGRAM:

1. Histogram is a graphical representation of the intensity distribution of an image.
2. In simple terms, it represents the number of pixels for each intensity value considered.

HISTOGRAM EQUALIZATION:

1. Histogram Equalization is a computer image processing technique used to improve contrast in images.
2. Histogram equalization is used to enhance contrast.
3. It is not necessary that contrast will always be increase in this.
4. There may be some cases were histogram equalization can be worse.
5. In that cases the contrast is decreased.
6. Let's start histogram equalization by taking this image below as a simple image as shown in figure 5.9

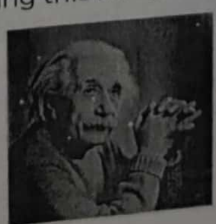


Figure 5.9

7. The histogram of this image has been shown below in figure 5.10.

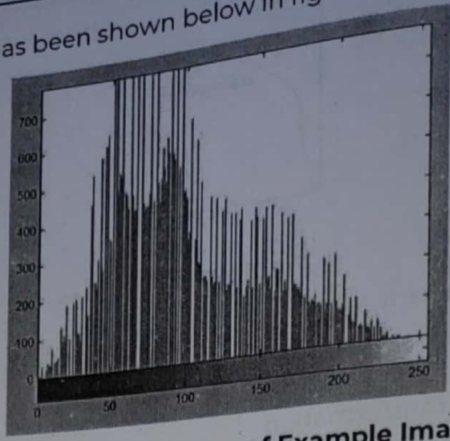


Figure 5.10: Histogram of Example Image.

8. Now we will perform histogram equalization to it.
9. First we have to calculate the PMF (probability mass function) of all the pixels in this image.
10. Next step is to calculate CDF (cumulative distributive function) according to gray levels.
11. Let's for instance consider this, that the CDF calculated looks like this.

Gray Level Value	CDF
0	0.11
1	0.22
2	0.55
3	0.66
4	0.77
5	0.88
6	0.99
7	1

12. Then in this step you will multiply the CDF value with (Gray levels (minus) 1) .
13. Considering we have a 3 bpp image.
14. Then number of levels we have are 8. And 1 subtracts 8 is 7. So we multiply CDF by 7.
15. Here what we got after multiplying.

Gray Level Value	CDF	CDF * (Levels-1)
0	0.11	0
1	0.22	1
2	0.55	3
3	0.66	4
4	0.77	5
5	0.88	6
6	0.99	6
7	1	7

16. In last step, we have to map the new gray level values into number of pixels.
17. Let's assume our old gray levels values has these number of pixels.

Gray Level Value	Frequency
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16

8. Now if we map our new values to, then this is what we got.

Gray Level Value	New Gray Level Value	Frequency
0	0	2
1	1	4
2	3	6
3	4	8
4	5	10
5	6	12
6	6	14
7	7	16

9. Now map these new values to the histogram, and we are done.
10. Figure 5.11 shows Histogram Equalization Image and figure 5.12 shows Histogram Equalization histogram.

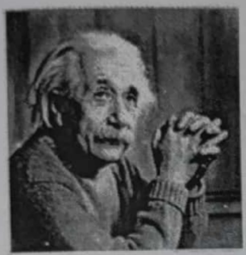


Figure 5.11: Histogram Equalization Image

Figure 5.12: Histogram Equalization histogram

Q5. Explain Histogram Stretching

[P | Medium]

Ans:

HISTOGRAM STRECHING:

1. Histograms are used for contrast enhancement.
2. There are two methods of enhancing contrast.
3. The first one is called Histogram stretching that increase contrast.
4. The second one is called Histogram equalization that enhance contrast.
5. Contrast is the difference between maximum and minimum pixel intensity.
6. Consider the image as shown in figure 5.13 and histogram of the image as shown in figure 5.14

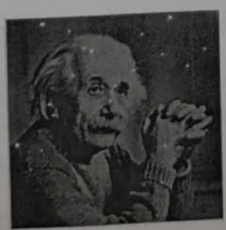


Figure 5.13: Image

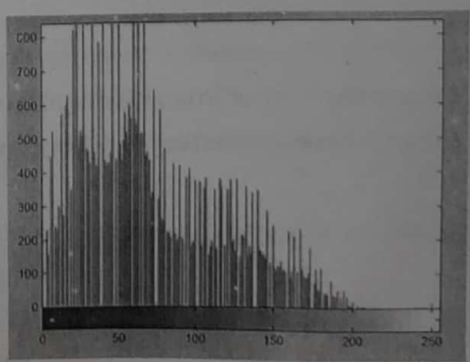


Figure 5.14: Histogram of the Image

7. Now we calculate contrast from this image. Contrast = 225.
8. The formula for stretching the histogram of the image to increase the contrast is

$$g(x,y) = \frac{f(x,y) - fmin}{fmax - fmin} * 2^{bpp}$$

9. The formula requires finding the minimum and maximum pixel intensity multiply by levels of gray.
10. In our case the image is 8bpp, so levels of gray are 256.
11. The minimum value is 0 and the maximum value is 225.
12. So the formula in our case is

$$g(x,y) = \frac{f(x,y) - 0}{225 - 0} * 255$$

13. where f(x,y) denotes the value of each pixel intensity.
14. For each f(x,y) in an image, we will calculate this formula.
15. After doing this, we will be able to enhance our contrast.

FAILING OF HISTOGRAM STRETCHING:

1. Histogram stretching algorithm fails on some cases.
2. Those cases include images with when there is pixel intensity 0 and 255 are present in the image
3. Because when pixel intensities 0 and 255 are present in an image, then in that case they become the minimum and maximum pixel intensity which ruins the formula like this.
4. Original Formula:

$$g(x,y) = \frac{f(x,y) - fmin}{fmax - fmin} * 2^{bpp}$$

5. Putting fail case values in the formula:

$$g(x,y) = \frac{f(x,y) - 0}{255 - 0} * 255$$

6. Simplify that expression gives

$$g(x,y) = \frac{f(x,y)}{255} * 255$$

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

7. That means the output image is equal to the processed image.
8. That means there is no effect of histogram stretching has been done at this image.

Q6. Explain Neighborhood Pixels Processing **[P | Medium]**

Ans:

NEIGHBORHOOD PIXELS PROCESSING:

1. It is also **spatial domain technique** in image enhancement.
2. Here, we consider one pixel at a time & modify it accordingly.
3. Its neighboring pixels are also taken in consideration.
4. So, we change pixel value based on 8 neighbors.
5. Along with 3x3 neighborhood, 5x5 & 7x7 can also be used.
6. A lot of things can be achieved by neighborhood processing which are not possible by point processing.

3 X 3 NEIGHBORHOOD / MASK / WINDOW / TEMPLATE:

	(y − 1)	y	(y + 1)
(x − 1)	W1 g(x − 1, y − 1)	W2 g(x − 1, y)	W3 g(x − 1, y + 1)
x	W4 g(x, y − 1)	W5 g(x, y)	W6 g(x, y + 1)
(x + 1)	W7 g(x + 1, y − 1)	W8 g(x + 1, y)	W9 g(x + 1, y + 1)

Figure 5.15: 3 X 3 Neighborhood / Mask / Window / Template

1. To achieve neighborhood processing, place the mask on the image.
2. Multiply each mask component with the pixel component.
3. Add them, place value at center. Similar to CONVOLUTION.
4. Only here we does not flip the mask as it is symmetric.
5. If g is original image & f is modified image, then:

$$f(x, y) = g(x-1, y-1).w1 + g(x-1, y).w2 + g(x-1, y+1).w3 + g(x, y-1).w4 + g(x, y).w5 + g(x, y+1).w6 + g(x+1, y-1).w7 + g(x+1, y).w8 + g(x+1, y+1).w9$$

6. Once f(x, y) is calculated, shift mask by 1 step to right.
7. Now, W5 coincide with g(x, y + 1).
8. One of the important operations that can be achieve using neighbourhood processing is that of image filtering
9. We can perform low pass filtering (LF), high pass filtering (HF) & band pass filtering using neighbourhood operations.
10. In 1D signals, if 2 signals represent voltage then, how fast the signal changes is indication of frequency.
11. Same concept is applied to images where we have gray levels instead.
12. If gray scale change slowly over a region then LF area. E.g. Background.
13. If gray scale change abruptly over a region then HF area. E.g. Edges, Boundaries.

Q7. Explain filtering in spatial domain

[P | Medium]

Ans:

SPATIAL FILTERING:

1. The use of spatial masks for image processing is called spatial filtering and the masks are called spatial filters.
2. Spatial filtering involves passing a weighted mask or kernel over the image and replacing the original image pixel value corresponding to the center of the kernel with the sum of original pixel values in the region corresponding to the kernel multiplied by the kernel weight.
3. For example consider digital sub-image F and 3 x 3 filter mask was given below:

   Input image F = $\begin{bmatrix} Z1 & Z2 & Z3 \\ Z4 & Z5 & Z6 \\ Z7 & Z8 & Z9 \end{bmatrix}$ and Filter Mask W = $\begin{bmatrix} W1 & W2 & W3 \\ W4 & W5 & W6 \\ W7 & W8 & W9 \end{bmatrix}$
4. The response of a linear mask is given as,

   $R = Z_1W_1 + Z_2W_2 + \cdots + Z_9W_9$
5. If the center of the mask is at location (x, y) in the image the gray level of the pixel located at (x, y) is replaced by R.
6. The mask is then moved to the next pixel location in the image and the process is repeated.
7. This continues until all pixel locations are covered.

TYPES OF FILTERS:

I) Smoothing Spatial Filters:

1. Smoothing filters are used for blurring and for noise reduction.
2. Blurring is used as preprocessing such as removal of small details from image.
3. Noise reduction is blurring with linear or non-linear filter.
4. Smoothing filter can be linear and non-linear.

a. Smoothing Linear Filter:

- The output of a smoothing linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.
- These filters sometimes are called **averaging filters or low pass filters.**
- It is used to remove noise present in the image.
- Noise is normally a high frequency signal and low pass filtering eliminates the noise.
- The major usage of average filter is reduction of irrelevant detail in an image.
- Figure 5.16 shows 3x3 smoothing filter masks

(1/9) x

1	1	1
1	1	1
1	1	1

(1/16) x

1	2	1
2	4	2
1	2	1

Figure 5.16: 3x3 smoothing filter masks

- Standard average of pixels calculated as follows:

$$R = (1/9)\sum_{i=1}^{9} Z_i$$

- At the end of filtering the entire image is divided by 9.

- So m x n is equal to 1/mn.
- Thus the coefficients pixels are equal, so the filter is called **box filter**.

b. Smoothing Non-Linear Filter:

- They are also known as **Ordered Statistic Filters**.
- Examples of Non-Linear filters are Median, Max and Min Filter.
- Median Filter is effective for impulse noise called as salt & pepper noise.
- Max Filters is use to find the brightest point of an image.
- Min Filters is use to find the darkest point of an image.

II) Sharpening Spatial Filters:

1. Sharpening is used to highlight fine detail in an image.
2. There are many applications, such as electronic priming, medical images, military systems that use sharpening technique.
3. Sharpening filters that are based on two derivatives. i.e. First Order Derivatives and Second Order Derivatives.

a. First Order Derivative:

- Must be zero for area of constant intensity.
- Must be nonzero of intensity step or map.
- Must be nonzero along ramp.
- First order derivative produce thick edge.
- It is given by $(\partial f/\partial x) = f(x + 1) - f(x)$
- Examples are Robert, Prewit, Sobel and Fri-Chen filter.

Robert

Gx

1	0
-1	0

Gy

1	-1
0	0

Sobel

Gx

-1	-2	-1
0	0	0
1	2	1

Gy

-1	0	1
-2	0	2
-1	0	1

Prewit

Gx

-1	-1	-1
0	0	0
1	1	1

Gy

-1	0	1
-1	0	1
-1	0	1

Fri-Chen

Gx

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

Gy

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

b. Second derivative:

- Must be zero in constant areas.
- Must be nonzero at one end and other end of intensity ramp.
- Must be zero along ramps.
- Second order derivative produce much finer one and strong response at isolated point.
- It is given by $(\partial^2 f/\partial x^2) = f(x + 1) - f(x - 1) - 2f(x)$
- Examples are Laplacian, High Pass and High Boost Filter.

Laplacian 4 Directional

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

Laplacian 8 Directional

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

High Pass Filter

1/9

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

High Boost Filter

1/9

-1	-1	-1
-1	9k - 1	-1
-1	-1	-1

CHAP - 6: IMAGE SEGMENTATION

Q1. Explain image segmentation in detail **[P | High]**

Ans:

IMAGE SEGMENTATION:

1. In computer vision, **image segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels).
2. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze.
3. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.
4. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

IMAGE SEGMENTATION BASED ON PROPERTIES:

Based on discontinuity:

1. In this approach, the image is partition based on abrupt changes in gray level.
2. The principle area of interest within this category is detection of isolated points and detection of lines and edges in an image.

I) Point Detection:

1. The detection of points is done by using following mask:

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

2. If |R| > T then isolated point is detected.
3. Where T is a non-negative point threshold. I.e. weighted difference between the center point and its neighbor.
4. $R = W_1Z_1 + W_2Z_2 + W_3Z_3 + \dots + W_9Z_9$
5. The idea is that gray level of an isolated image will be quite different from gray level of its neighbors.

II) Line Detection:

1. The various mask present for line detection are:

Horizontal

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

+45°

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

Vertical

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

-45°

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

2. The first mask respond more strongly to line oriented horizontally.
3. While third mask respond more strongly to line oriented vertically and 2nd and 4th mash mask respond more strongly to line oriented + 45 and – 45 degrees.
4. With constant background the maximum response would result when the line is passing through the middle row of the mask.

III) Edge Detection:

1. It is the most common approach for detection of discontinuities in grey level.
2. Edge Detection characterizes the object boundaries.
3. Edge point can be thought of as pixel location of abrupt gray levels.
4. It is the boundary between two regions with relatively distinct gray levels properties.
5. There are two types of edges. They are step and ramp edge.
6. Step edges are detected using first order derivative filters like Robert, sobel, frichen and prewit.
7. Ramp edges are detected using second order derivative line Laplacian filter.

Based on similarity:

1. In this approach the principal approach is threshold, region growing and region splitting and merging.
2. In this approach, segmentation is done by finding the regions directly.
3. Let R represents the entire image region segmentation as a process that partitions R into 'n' sub regions R1, R2, R3 ... Rn such that,
   a. R1 U R2 U .... U Rn = R
   b. Ri is connected region where i = 1, 2, 3 ... N
   c. Ri $\cap$ rj = $\varnothing$ for i = j
   d. Predicate (Ri) = True for all i = 1, 2, ... N
4. It can be applied to both static as well as dynamic images.
5. There are two different approaches for region oriented segmentation.

I) Region growing by pixel aggregation:

1. It is the procedure that groups pixels or sub- regions into larger regions.
2. Pixel aggregation procedure starts with a set of seed point and from these grows region by appending for each seed point those neighboring pixels that have similar proportion.
3. Region growing is better than edge detection technique in noisy images where are edges are difficult to detect.

Algorithm:

a. Let R be a region to extract. (Initially the region R only contains the seed point p)
b. Let F be a FIFO that only contain a boundary point R. (Initially F contains 8 neighborhood of seed point p)
c. While F is not empty
d. For each neighbor pixel P* of a P in F
e. If P* is similar to P
f. Then P is added to R & Neighbor pixel of P* is added to F
g. Else set P* as non-similar

II) Region spitting & merging:

1. In this method the image is first divided into a set of arbitrary disjoint region and then merges and/ or splits the regions.
2. Let R represent the entire image region and then select a predicate P.
3. Now sub divide the regions into smaller and smaller quadrant region so that for any region Ri, predicate (Ri) = true.
4. If Predicate (region) = false then divide the image into quadrants.

5. If Predicate (region) = false for any quadrant then subdivide that quadrant into sub-quadrants and so on.

Algorithm:

a. Split into four disjointed quadrants any region Ri when predicate (Ri) = false.
b. Merge any adjacent regions Rj & Rk for which predicate (Rj U Rk) = true.
c. Stop when NO further merging or splitting is possible.

Q2. Explain image edge detection using Previtt Mask [P | Medium]

Ans:

IMAGE EDGE DETECTION USING PREVITT MASK:

1. It is one of the method of gradient edge detection methods.
2. The Prewitt operator is used for detecting vertical and horizontal edges in images.
3. Edges are calculated by using difference between corresponding pixel intensities of an image.
4. All the masks that are used for edge detection are also known as derivative masks.
5. All the derivative masks should have the following properties:
   a. Opposite sign should be present in the mask.
   b. Sum of mask should be equal to zero.
   c. More weight means more edge detection.
6. Prewitt operator provides us two masks one for detecting edges in horizontal direction and another for detecting edges in a vertical direction.

Vertical Direction:

-1	0	1
-1	0	1
-1	0	1

1. Above mask will find the edges in vertical direction and it is because the zeros column in the vertical direction.
2. When you will convolve this mask on an image, it will give you the vertical edges in an image.
3. When we apply this mask on the image it prominent vertical edges.
4. It simply works like as first order derivate and calculates the difference of pixel intensities in a edge region.
5. As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right and left pixel values around that edge.
6. This increase the edge intensity and it becomes enhanced comparatively to the original image.

Horizontal Direction:

-1	-1	-1
0	0	0
1	1	1

1. Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction.
2. When you will convolve this mask onto an image it would prominent horizontal edges in the image.
3. This mask will prominent the horizontal edges in an image.
4. It also works on the principle of above mask and calculates difference among the pixel intensities of a particular edge.
5. As the center row of mask is consist of zeros so it does not include the original values of edge in the image but rather it calculate the difference of above and below pixel intensities of the particular edge.
6. Thus increasing the sudden change of intensities and making the edge more visible.
7. Both the above masks follow the principle of derivate mask.
8. Both masks have opposite sign in them and both masks sum equals to zero.
9. The third condition will not be applicable in this operator as both the above masks are standardize and we can't change the value in them.

Q2. Explain image edge detection using Sobel

Ans: **[P | Medium]**

IMAGE EDGE DETECTION USING SOBEL OPERATOR:

1. It is one of the method of gradient edge detection methods.
2. The sobel operator is very similar to Prewitt operator.
3. It is also a derivate mask.
4. Like Prewitt operator, sobel operator is also used to detect two kinds of edges in an image:
   a. Vertical Direction
   b. Horizontal Direction
5. The major difference is that in sobel operator the coefficients of masks are not fixed and they can be adjusted according to our requirement unless they do not violate any property of derivative masks.

Vertical Mask of Sobel Operator:

-1	0	1
-2	0	2
-1	0	1

1. This mask works exactly same as the Prewitt operator vertical mask.
2. There is only one difference that is it has "2" and "-2" values in center of first and third column.
3. When applied on an image this mask will highlight the vertical edges.
4. When we apply this mask on the image it prominent vertical edges.
5. It simply works like as first order derivate and calculates the difference of pixel intensities in a edge region.
6. As the center column is of zero so it does not include the original values of an image but rather it calculates the difference of right and left pixel values around that edge.
7. Also the center values of both the first and third column is 2 and -2 respectively.
8. This give more weight age to the pixel values around the edge region.

9. This increase the edge intensity and it becomes enhanced comparatively to the original image.

Horizontal Mask of Sobel Operator:

-1	-2	-1
0	0	0
1	2	1

1. Above mask will find edges in horizontal direction and it is because that zeros column is in horizontal direction.
2. When you will convolve this mask onto an image it would prominent horizontal edges in the image. The only difference between it is that it have 2 and -2 as a center element of first and third row.
3. This mask will prominent the horizontal edges in an image.
4. It also works on the principle of above mask and calculates difference among the pixel intensities of a particular edge.
5. As the center row of mask is consist of zeros so it does not include the original values of edge in the image but rather it calculate the difference of above and below pixel intensities of the particular edge.
6. Thus increasing the sudden change of intensities and making the edge more visible.

Q3. Explain image edge detection using Laplacian Operator

Ans:

[P | Medium]

IMAGE EDGE DETECTION USING LAPLACIAN OPERATOR:

1. Laplacian Operator is also a derivative operator which is used to find edges in an image.
2. The major difference between Laplacian and other operators like Prewitt, Sobel, Robinson and Kirsch is that these all are first order derivative masks but Laplacian is a second order derivative mask.
3. In this mask we have two further classifications one is Positive Laplacian Operator and other is Negative Laplacian Operator.
4. Another difference between Laplacian and other operators is that unlike other operators Laplacian didn't take out edges in any particular direction but it take out edges in following classification.
   a. Inward Edges
   b. Outward Edges

Positive Laplacian Operator:

1. In Positive Laplacian we have standard mask in which center element of the mask should be negative and corner elements of mask should be zero.

0	1	0
1	-4	1
0	1	0

2. Positive Laplacian Operator is use to take out outward edges in an image.

Negative Laplacian Operator:

1. In negative Laplacian operator we also have a standard mask, in which center element should be positive.
2. All the elements in the corner should be zero and rest of all the elements in the mask should be -1.

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

3. Negative Laplacian operator is use to take out inward edges in an image

WORKING:

1. Laplacian is a derivative operator; its uses highlight gray level discontinuities in an image and try to deemphasize regions with slowly varying gray levels.
2. This operation in result produces such images which have grayish edge lines and other discontinuities on a dark background.
3. This produces inward and outward edges in an image
4. The important thing is how to apply these filters onto image.
5. Remember we can't apply both the positive and negative Laplacian operator on the same image.
6. We have to apply just one but the thing to remember is that if we apply positive Laplacian operator on the image then we subtract the resultant image from the original image to get the sharpened image.
7. Similarly if we apply negative Laplacian operator then we have to add the resultant image onto original image to get the sharpened image.

Join **BackkBenchers Community** & become the **Student Ambassador** to represent your college & earn 15% Discount.

Be the **Technical Content Writer** with BackkBenchers and earn upto 100 Rs. per 10 Marks Questions.

Buy & Sell **Final Year Projects** with BackkBenchers. Project Charge upto 10,000.

Follow us on **Social Media Profiles** to get notified

BackkBenchersCommunity **+91-9930038388** **BackkBenchersCommunity**

E-Solutions Will be Available @BackkBenchers Website Soon.