IP ** Important, Some Theory Questions

Q). Why median filter is better than ~~to~~ mean filter?

Ans Median filter is normally used to reduce noise in an image, similer to the mean filter. However, it often does a better job than mean filter in preserving useful details in an image.

Median filter has 2 main Advantages.

1. The median is a more robust average than the 'mean' and so a single very un-representative pixel in a neighborhood will not affect the median value significantly.

2. Since the median value must actally be the value of one of the pixel in the neighborhood the median filter does not create new unrealistic pixel values when the filter straddles an edge. Therefore, it is much better at preserving 'sharp edges' than the mean filter.

#Note:- Mean filter is better at dealing with Gaussian noise than median filter.
- Median filter is better at dealing with salt and pepper noise than mean filters.

Write down a few approaches to deal with missing edge pixels.

** These are Missing edge pixel

eg:-

| | | | |
|---|---|---|---|
| 50 | 50 | 50 | 60 |
| 50 | 50 | 50 | 60 |
| 50 | 60 | 50 | 50 |
| 40 | 50 | 60 | 50 |

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

A few approaches to dealing with missing edge pixels are :-

1) Omit missing pixels.
   - only works with some filters.
   - Can add extra code and slow down processing.

2) Pad the image (eg: 0).
   - Typically with either all white or black pixels.

3) Replicate Border pixel :-

4) Truncate the image.

5) Allow pixels used around the image
   - Can cause some strange image artifacts

Ex:- ②

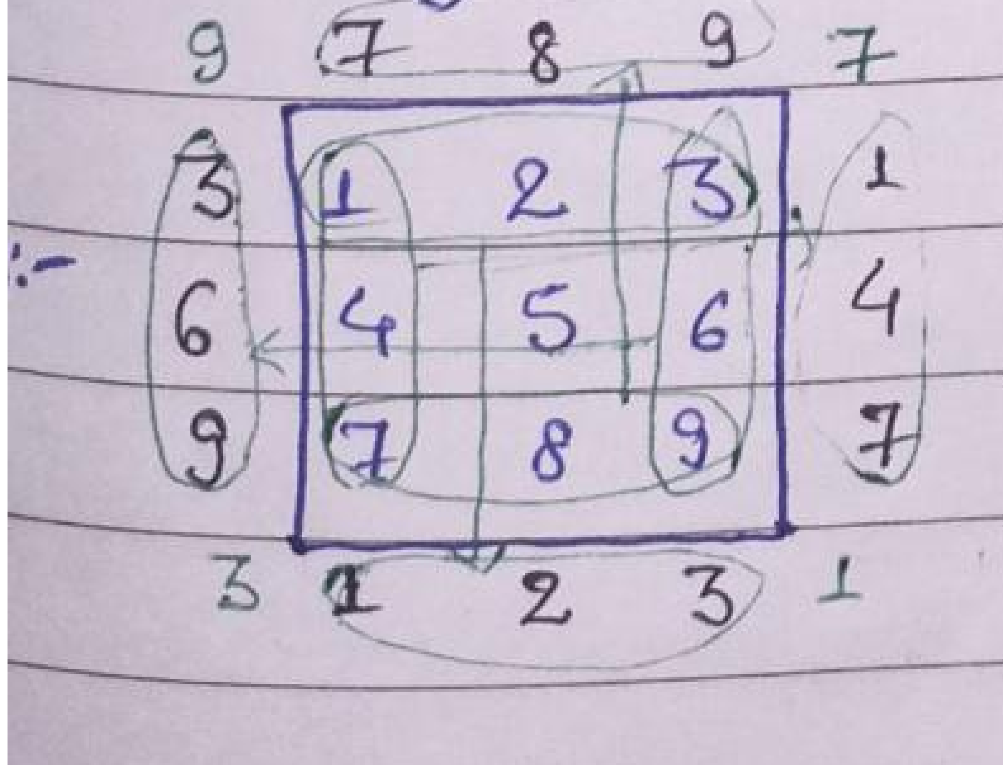
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 4 | 8 | 0 |
| 0 | 5 | 4 | 0 | 0 |
| 0 | 5 | 6 | 9 | 0 |
| 0 | 0 | 0 | 0 | |

③

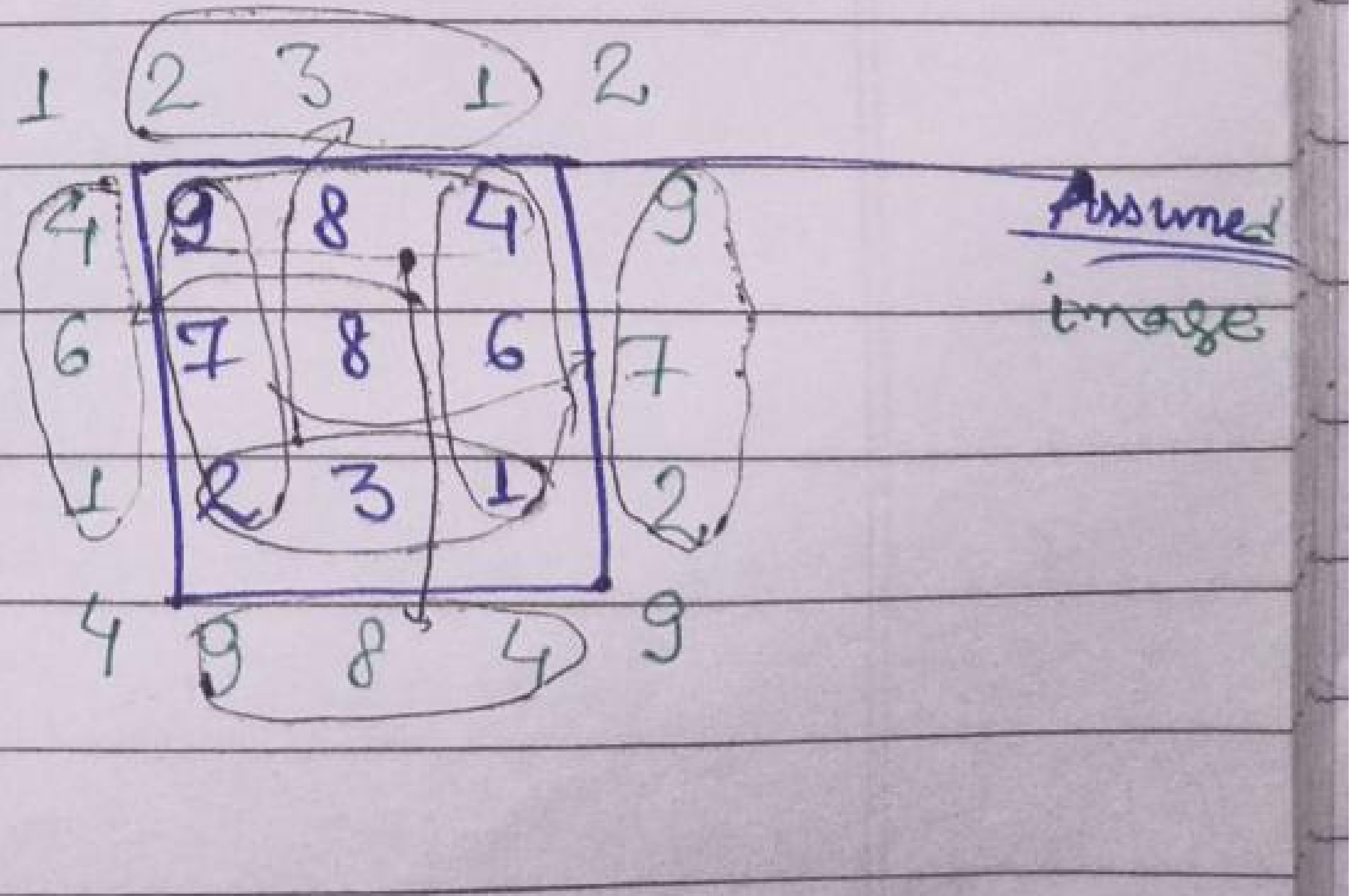
| | | | | |
|---|---|---|---|---|
| 1 | 1 | 2 | 3 | |
| 1 | 1 | 2 | 3 | 3 |
| 4 | 4 | 5 | 6 | 6 |
| 7 | 7 | 8 | 6 | 6 |
| 7 | 7 | 8 | 6 | 6 |

ex:- replicate

Wrapping of Pixels.:-



2nd exam:-



IP

# Sharpening Spartial filter

PAGE NO. :
DATE : 19. 08. 2

⇒ The principal objective of sharpening is to highlight Transitions in intensity.

* Blurring → Pixel averaging.
* Sharpening → Spartial Differentiation

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

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

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

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

* Laplacian filter

## Examples

Q) ① Apply Laplacian filter on the given image on the center pixel.

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

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

$= 0+5+0+0-24+2+0+3+0$

$= 10-24 = \boxed{-14}$ will placed at ⑥ cent.

* Enhanced laplacian Filter :-

We just increment with '1'.

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

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

Q)② Apply enhanced Laplacian filter on the given image on the center pixel.

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

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

$= \cancel{30}-5 \quad 8+5+4+0$

$\textcircled{-54}+2+1+3+7$

$= 30-54 = \boxed{-24}$ It will substitute of center value.

Q(3) Apply Laplacian filter on the given image :-

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 50, | 50, | 50, | 50 | 100 | 100 | 100 | 100 |
| 50, | 50-8 | 50, | 50 | 100 | 100 | 100 | 100 |
| 50, | 50. | 50, | 50 | 100 | 100 | 100 | 100 |
| 50 | 50 | 50 | 50 | 100 | 100 | 100 | 100 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |
| 100 | 100 | 100 | 100 | 50 | 50 | 50 | 50 |

mask →

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

Iteration :-

(1) $= \cancel{50\times 8} - \cancel{80}\, 50\times 8$

$= 0$

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

$= 450 + 300 - 400 = 150.$

(4) $50\times 3 + 100\times 5 - 400$

$= 150 + 500 - 400 = -150$

Ans

Final:-

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

Ans

# Un sharp Masking & High boost (Filter)

## * Un Sharp Masking :-

$$f_s(x, y) = f(x, y) - \bar{f}(x, y).$$

sharpened image = original image − blurred image.

* Subtracting a blurred version of an image from the original produces a sharpened image.

## * Highboast filtering :-

$$f_{hb}(x, y) = A f(x, y) - \bar{f}(x, y)$$

$$= A f(x, y) - [f(x, y) - f_s(x, y)]$$

$$\boxed{f_{hb}(x, y) = (A-1) f(x, y) - f_s(x, y).}$$

This is the generalized form of ~~to~~ Unsharp masking where $\boxed{A \geq 1}$, A Specifies the amount of sharpening of the image.

⇒ If we use Laplacian filter to create the sharpened image $f_s(x, y)$ with addition of the original image :-

$$f_s(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y) \\ f(x,y) + \nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f(x,y) \\ Af(x,y) + \nabla^2 f(x,y). \end{cases}$$

* Highboost Mask :-

Q. Imp

| | | |
|---|---|---|
| 0 | -1 | 0 |
| -1 | A+4 | -1 |
| 0 | -1 | 0 |

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | A+8 | -1 |
| -1 | -1 | -1 |

where $A \geq 1$

Q. Apply highboost filter on the image given below on the Center pixel. use the Mask with $A = \boxed{1.7}$

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

I/p image

$*$

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | A+8 | -1 |
| -1 | -1 | -1 |

$$= -1(1+2+3+4+6+7+8+9) + 5(1.7+8)$$

$$= -1(40) + 5(9.7)$$

$$(-40 + 48.5) = 8.5$$

This $\boxed{8.5}$ will Replace to $\boxed{5}$ that will be Ans

# First-Order Derivative Filter

1. **Robert operators** (~~Gr~~ Cross-gradient)

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

* It used to ~~ero~~ cross diagonal differentiation.

$$\begin{bmatrix} 5 & 6 & 8 \\ 0 & \boxed{25} & 2 \\ 9 & 1 & \boxed{5} \end{bmatrix} \text{ Ex.}$$

▷ Problems with Robert Cross :-

1. 2x2 Mask are not easy to implement
2. No. of calculations are More
3. No. of neighboring pixels considered in one go are less

⇒ To solve these Problems, we Make the following Changes :- 1). change in the size of the Mask
2). change in the No. of neighboring Pixels considered

* **Sobel operators** :-

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & \boxed{0} & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & \boxed{0} & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

* **Prewitt operators** :-

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Q.2 Apply Roberts, sobel and prewitt operators on the pixel (1, 1) int the following image.

$$\begin{array}{|ccc|c} 50 & 50 & 100 & 100 \\ 50 & \textcircled{50} & 100 & 100 \\ 50 & 50 & \textcircled{100} & 100 \\ \hline 50 & 50 & 100 & 100 \end{array}$$

i/p Image

1). Roberts operators (2x2)

origin

$$\begin{array}{|cc|} \hline \textcircled{-1} & 0 \\ 0 & \textcircled{1} \\ \hline \end{array}$$

$= 50 \times (-1) + 100 \times 1$

$= -50 + 100 = \boxed{50}$

2). sobel operator.

$$\begin{array}{|ccc|} \hline -1 & -2 & -1 \\ 0 & \textcircled{0} & 0 \\ 1 & 2 & 1 \\ \hline \end{array}$$

$= 50(-1) + 50(-2) + 100(-1) + 50(1) + 50(2) + 100(1)$

$= \cancel{-50} \cancel{-100} \cancel{-100} \cancel{+50} \cancel{+100} \cancel{+100} = \boxed{0}$

3). Prewitt operator :-

$$\begin{array}{|ccc|c} 50 & 50 & 100 & 100 \\ 50 & \textcircled{50} & 100 & 100 \\ 50 & 50 & 100 & 100 \\ \hline 50 & 50 & 100 & 100 \end{array} * \begin{array}{|ccc|} \hline -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ \hline \end{array}$$

I/p image

$= -1(50 + 50 + 100) + 1(50 + 50 + 100).$

$= \boxed{0}$ it will assigned at center $\textcircled{50}$ Answer.

DFT

PAGE NO.:
DATE: 18 / 08 / 2

Q. 1 Compute DFT of the sequence $f(x) = \{\underset{0}{1}, \underset{1}{0}, \underset{2}{0}, \underset{3}{1}\}$

$$F[k] = \sum_{x=0}^{N-1} f(x) e^{-j2\pi k x/N} \text{ ; Where } k = 0, 1, 2 \dots N-1.$$

Here $x = 0$ ; $N = 4$

$$F[k] = \sum_{x=0}^{3} f(x) e^{-j2\pi k x/4}$$

$$= f(0)e^{0} + \underset{\downarrow}{\boxed{x=1}}\; f(1)e^{-j2\pi k/4} + \underset{\downarrow}{\boxed{x=2}}\; f(2)e^{-j\pi k} + \underset{\downarrow}{\boxed{x=3}}\; f(3)e^{-j3\pi k/2}$$

$$= 1 + 0 + 0 + f(3)e^{-j3\pi k/2}$$

$$= \boxed{1 + e^{-j3\pi k/2}}$$

When $k = 0$

$F[0] = 1 + e^{0} = 1 + 1 = \boxed{2}$

When $k = 1$
$F[1]$ $= 1 + e^{\cancel{-j3\pi k/4}\; -j3\pi/2} = 1 + j$

When $k = 2$
$F[2]$ $= 1 + e^{-j3\pi} = 1 - 1 = 0$

When $k = 3$
$F[3]$ $= 1 + e^{-j9\pi/2} = 1 - j$

* $F[k] = \{2, 1+j, 0, 1-j\}$ Final Answer.

** Exam

we know, * Kernel of a 4-point DFT

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \rightarrow \text{Kernel}$$

* 1D DFT : $F[k] = \text{Kernal} \times f(x)$.

* 2D DFT : $F[k,l] = \text{Kernal} \times f(x,y) \times \text{Kernel}^T$ (Transpose)

Ques: Calculate 4-point DFT ~~in one~~ for the sequence $x(n) = \{0, 1, 2, 3\}$ using matrix Method.

Ans The 4-Point DFT in one dimensional = Kernel × Input Sequence.

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 0 + 1 + 2 + 3 \\ 0 - j - 2 + 3j \\ 0 - 1 + 2 - 3 \\ 0 + j - 2 - 3j \end{bmatrix} = \begin{bmatrix} 6 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$ Ans

IP

** Question / Answers

Q). ③ Compute the 2D DFT of the gray scale image is given by.
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Ans $F(k,l) = \boxed{\text{Kernal} \times f(m,n) \times \text{Kernal}^T}$

Kernal for the 4-point DFT :-
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$F[k,l] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \times \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}}_{\text{Given img}}$$

$$(\text{Kernal}^T) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$F[k,l] = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

4+4+4+4 4+4j+4[illegible]

4+4+4+4. ~~4+~~ (4×j + ~~-4+4~~[illegible]
=0.

$$F(k,l) = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Ans

Scanned by TapScanner

## * Discrete Cosine Transform :-

- Expresses a finite sequence of datapoints in terms of a sum of cosine functions oscillating at different frequencies.

⇒ Represents an image as a sum of ~~the~~ sinusoids of varying magnitudes of frequencies.

## * One Dimensional DCT

$$X[k] = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left\{\frac{(2n+1)\pi k}{2N}\right\}$$

where $0 \le k \le N-1$;

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } k=0 \\ \sqrt{\frac{2}{N}}, & \text{if } k \neq 0 \end{cases}$$

Practise

## *Bases or Kernel of a 4-point DCT *

$$= \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6532 & 0.6533 & -0.2706 \end{bmatrix} \text{(06)}$$

▷ 1-D DCT :- $F[k] = \text{Kernel} \times f(x)$

▷ 2-D DCT :- $F[k, l] = [\text{Kernel} \times f(x, y) \times \text{Kernel}^T]$

Questions ⇓

IP

1-D

Ques ex:- Find the DCT of $f(x) = (1, 2, 4, 7)$

Sol^n $F$ = Basis function × $f(x)$.

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -05 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 4 \\ 7 \end{bmatrix}$$

$$= \begin{bmatrix} 7 \\ -4.45 \\ 1 \\ -0.370 \end{bmatrix}$$ Ans

Q). ex) Find 2-D DCT of $f(x,y) =$

$$(x,y) = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix}$$

Know

Sol^n we have:-

$F$ = Kernel × $f(x,y)$ × Kernel$^T$

$$F = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} 0.5 & 0.6532 & 0.5 & 0.2706 \\ 0.5 & 0.2706 & -0.5 & -0.6533 \\ 0.5 & -0.2706 & -0.5 & 0.633 \\ 0.5 & -0.6532 & 0.5 & -0.2706 \end{bmatrix}^T =$$