DIP-1

- imread() • imshow() • figure() subplot(221)
- Colormap in DIP – Changes color of grayscale Images

SOME GEN. INFO

- An Image in MATLAB is rep as a Matrix of pixel values
- Each pixel → value → value of pixel determines color
- Color of a pixel is determined by the colormap that is associated with the Image

colormap = 'Map' the Pixel values of an Image TO THE Colors in a Colormap – returns vector of [r, g, b] – row value depends on scaled value.

$\boxed{m \times 3}$ of no. B/w 0.0 and 1.0

each row = distinct color | 3 colomns → R-G-B only or any comb^n NO

STEPS

1] The Pixel values are scalled B/w 1 and m, where m is the no. of colors in colormap.

$$\text{scaled value} = \frac{[\text{value} - \text{min value}] * m}{[\text{maxvalue} - \text{minvalue}]}$$

- minvalue = minimum Pixel value of Image
- max value = maximum Pixel value of Image

2] Scaled values are mapped to colors in colormap matrix. Done by Interpolating B/w the colors defined by colormap

Interpolation → estimating a value that falls B/w 2 known values.

Eg Pixel values → $\boxed{0-255}$ and m = 5 with 3 cols → RYB and value = 100

- scaled value $= \dfrac{5[100-0]}{255} = \boxed{1.96}$ – B/w 1 and 2
- B/w Red and Yellow = ORANGE or Shade of orange

- Repeat the steps for all pixels and colors are associated to each scalled Pixel values

Eg → m=5 - so matrix (5×3) colormap $\boxed{\text{matrix}}$

colormap returns the vector [r, g, b] the row value is decided by the INTERPOLATED VALUE OF SCALED PIXEL VALUE B/w 1 and m. $\boxed{\uparrow \text{working of colormap}}$ ↑

$\boxed{\text{Example}}$ Image pixel value [0-255]
colormap matrix [5×3] →
pixel value = 100

	R	G	B
R/1			
Y/2			
B/3			
G/4			
O/5			

[r, g, b] [r+g+b]

→ scaled value $= \dfrac{5(100-0)}{255} = \boxed{1.96}$

1.96 lies B/w 1 and 2. - Interpolation is used to estimate the value and corresponding [r, g, b] vector is selected as the color for that pixel from the colormap.

———— X ———— X ———— X ——— X ———

Interpolation : estimating a value that falls B/w 2 known values.

Linear Interpolation : uses straight line B/w 2 known points
Cubic spline Interpolation : fitting smooth curve to the known data points
Polynomial Interpolation : fitting polynomial curve B/w data points.

———— X ———— X ———— X ——— X ———

Linear Interpolation → (x_1, y_1) and (x_2, y_2) - LI of x →

$$y = y_1 + (x - x_1) * \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

Y of LI of x can be discovered by

———— X ———— X ———— X ———

Polynomial Interpolation → $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + .. + a_n x^n$

$\boxed{a_0, a_1, a_2, a_3 .., a_n}$ calc. by $\boxed{\text{sys of eq}}$ based on known Data Points $(x_1, y_1) \dots (x_n, y_n)$

COLOR MODELS

1] RGB → **ADDITIVE COLOR MODEL** in which colors are created by combining Diff levels of Red-Green-Blue

- (0-255) of 3 values
- Red = (255, 0, 0) - Yellow = (255, 255, 0)

———x———x———x———x———x———

2] CMYK → **SUBTRACTIVE COLOR MODEL**. In which colors are stored/created by subtracting Different values/levels of Primary colors → CYAN-MAGENTA-YELLOW-BLACK from WHITE

- (0-100) of 4 values
- Red = (0, 100, 100, 0) - Yellow = (0, 0, 100, 0)

———x———x———x———x———x———

3] HSL (Hue - Saturation - Lightness) - Colors are Represented as comb^n of Hue - Saturation and Lightness.

Hue → Dominant wavelength of light that is perceived by the Human eye.
- Rep as Angle on a Color wheel - 0° = red | 120° = green | 240° = blue

Saturation → Amount of color present in particular Hue.
- Rep as % - 0% = shade of gray | 100% = pure Vibrant color

Lightness → relative Brightness.
- rep as % - 0% = Black | 100% = White | 50% = Normal color

HSV (Hue - Saturation - Value)

Value/Brightness - luminosity of image/color

0% = Black | 100% = White

- Red : HSL (0°, 100%, 50%) , HSV (0°, 100%, 100%)
- Purple : HSL (300°, 100%, 50%), HSV (300°, 100%, 100%)

—— x —— x —— x —— x ——

4] LAB (Lab*) - Colors as combn of lightness and 2 color-opponent dimensions, a and b.

- L* → lightness of a color – 0 = Black | 100 = White
- a* → color – opponent dimension ranges from green to Red
  - +ve value = shades of red
  - -ve value = shades of green
- b* → color - opponent dimension ranges from Blue to Yellow
  - +ve value = shades of Yellow
  - -ve value = shades of Blue

- Red : Lab* (53.23, 80.11, 67.22)
- Purple : Lab* (60.32, 98.25, - 60.94)

—— x —— x —— x ——

5] YIQ - Color model used in National Television System Committee

- Colors as combn of luminance (Brightness) Y and Chrominance / Information (I and Q)

(Y, I, Q)

- $Y = 0.299R + 0.587G + 0.114B$
- $I = 0.596R - 0.274G - 0.322B$
- $Q = 0.211R - 0.523G + 0.312B$

* allows chrominance and luminance of an Image to be encoded and Transmitted seperately. - long distance - ↑ or ↔ quality.

—— x —— x —— x ——

6] YUV - Y → luminance

$Y = 0.299R + 0.587G + 0.114B$

U and V = Chrominance (Y,U,V)

→ $U = -0.147R - 0.289G + 0.436B$

$V = 0.615R - 0.515G - 0.100B$

---

7) Grayscale → Colors as shades of gray. Rep B/W

• Each pixel value is represented by a single value that Rep the Brightness of the pixel. — 8 Bit Integer — 0 = Black | 255 = white. Intermediate values = Shades of Gray.

↑ MAIN

* Grayscale value = $\boxed{(R+G+B)/3.}$

• Higher Intermediate Value = Lighter shades
• lower Intermediate value = Darker shades

* luminosity formula => $\boxed{\text{Grayscale}} = \boxed{0.21R + 0.72G + 0.07B}$

• Black - 0	Dark gray - 100
• white - 255	
• Light gray - 200	

Indexed Images

Type of Image Representation in which the Pixel Values in the Image are → Indices into a seperate color map or Table pallte.

Colormap specifies the actual colors that correspond to each Index values in the Image.

An Indexed Image is stored with a limited no. of colors (256 or fewer) — makes it more efficient to store and process in comparison to a full-color image.
↑ Easy Storage

Index Images are less versatile and can only Display the colors present in the colormap ← Limit to colors

- Used in GIF and PNG.

Example → Indexed Image → $\begin{bmatrix} 1 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 \end{bmatrix}$ — Storing Index values

RGB

↓ mapping

Index: 0 1 2

Color: (0,0,0) (255,0,0) (0,255,0)

X —— X —— X

Labelled Images

- Images that have been annotated with labels or Tags that Identify the objects or features present in the image.

X —— X —— X —— X

HDR - High Dynamic Range

luminosity → Measure of amount of light that is emitted by an object or scence.

↓

Units → luminous flux → Amount of light energy emmitted per unit of Time.

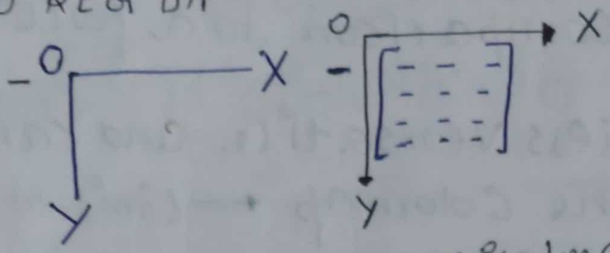
HDR → wider Range of luminosity in comparison to standard digital Imaging.

- Created By combining Multiple Images taken at different exposures into a single image.

DIP-2

SOME GENERAL INFO REG DIP

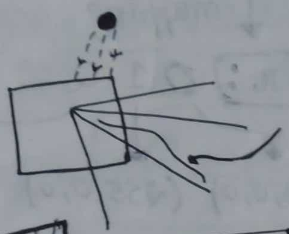
- Axis is Inverted –



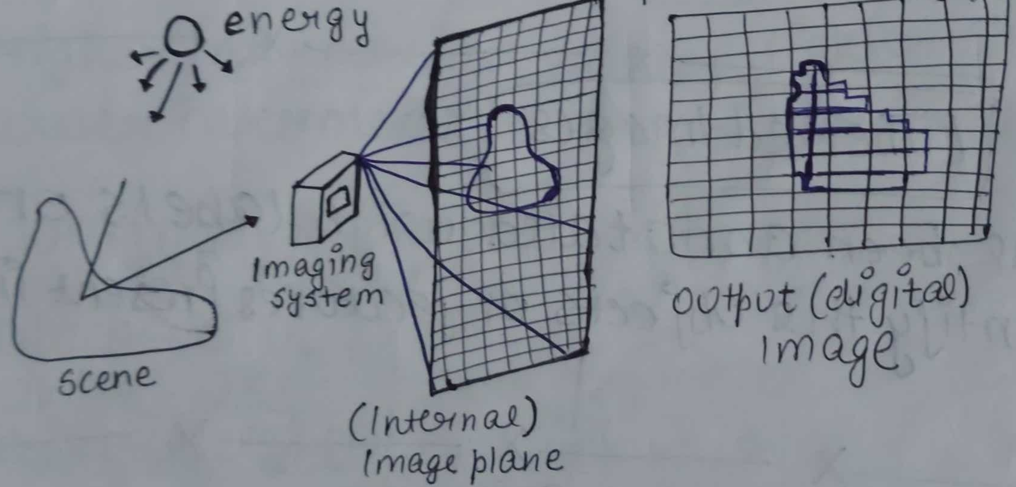
→ $f(x,y) = (x,y), (P_1,P_2)$

- Matrix Associated with Black and white Images – **BITMAP**
- Matrix Associated with Greyscale Images – Gray MAP
- Values 255 are Just for Representation, They are stored in Binary Format.

Perspective Projection →



distance (D) → D↑ size↓



* Capturing (x, y) → Display (m, n) ∴ $m > x$, $n > y$

L fill values in m and n on its own
　L causes Blur
　　L losses color and in continous Display movement / involvment loses coordinate val.

* $\boxed{0 \leq f(x,y) \leq \infty}$

DIP - 3

Changing Constrast / Increasing → histeq()

viewing histogram distribution → Imhist()

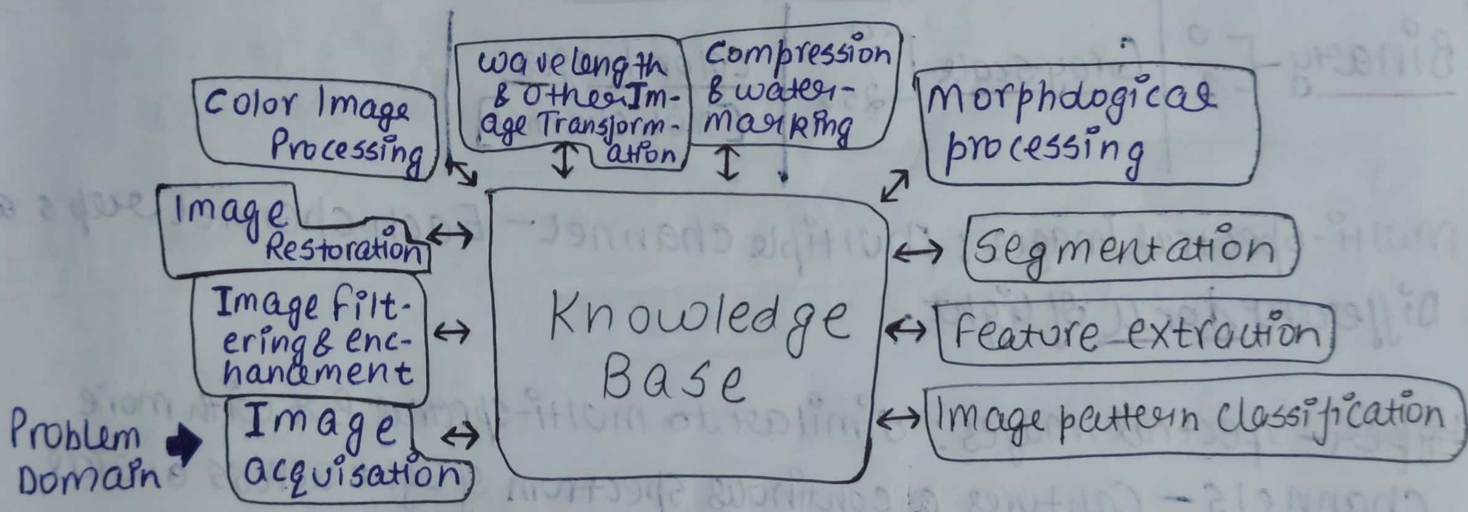
Contrast Refers to the Range of colors B/w lightest and darkest areas of the Image.

High Contrast large Difference B/w lighest and darkest areas

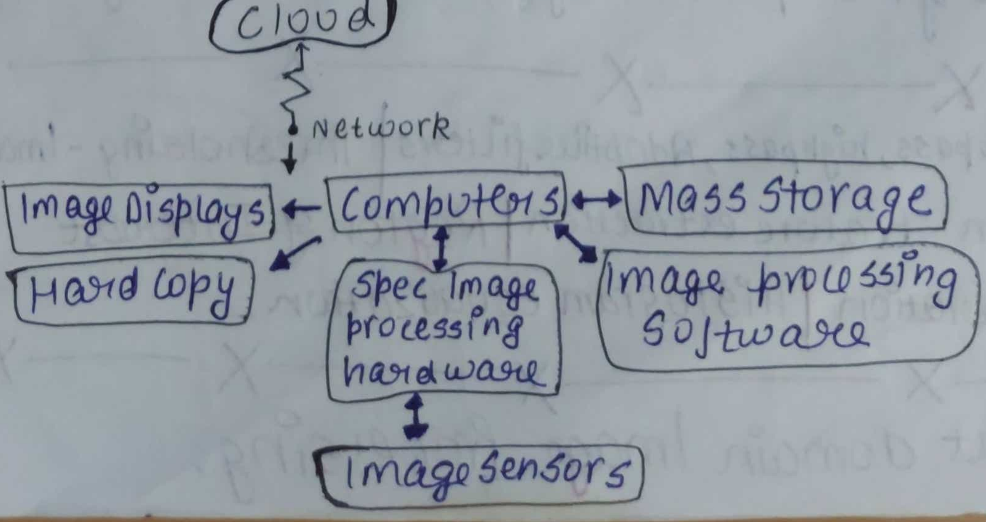
low Constrast small difference B/w lighest and darkest areas

* $\text{greyscale} = (R+G+B)/3$

Fundamental Steps in Digital IP



COMPONENTS OF A GENERAL PURPOSE IMAGE PROCESSING SYSTEM



* Function $f(x,y)$ is characterized by 2 components
 - ① Amount of source illumination INCIDENT on the scene
 - ② Amount of illumination REFLECTED by the Object

① - Illumination component - $i(x,y)$
② - reflectance component - $r(x,y)$

$$\boxed{f(x,y) = i(x,y)\, r(x,y)} \quad \text{—— (1)}$$

* $0 \le i(x,y) < \infty$ | * $0 \le r(x,y) \le 1$

—— X —— X —— X —— X ——

TYPE OF IMAGES

Binary - [0, 1] | **Gray scale** - [0, 255] | **Color Images** [Mul channel]

Multi-spectral Images : Multiple channel - Each channel reps a Different λ or μ of light.

Hyper-spectral Images : similar to multi-spectral But with more channels - Captures a continous spectrum of light across a wide Range of λ or μ.

Volumetric Images : 3D representation of an Object and it consists of multiple 2D Images Stacked on Top of each Other.

—— X —— X —— X —— X ——

Filters - low pass, high pass, Adaptive filters | Thresholding - Image Segmentation
Edge detection - feature extraction | Region of Interest
Bit plane Operation | Histogram equalization

—— X —— X —— X —— X —— X ——

Gradient domain Image Processing.

Cont Analog Images into Digital form

DIP-4

Ch 2-END

- Image Sampling and Quantization

- Scan line
  - Compromise in Color - Quantization
  - compromise in coor value - sampling.
  - $V = \{mul\ col\}$
    - affects m-adjacency

- Rel^n B/w Pixels
  - 3 kinds of neighbors of a pixel {4, 8, Diagonal}
  - Adjancency (3 types) {4, 8, [redacted] m}

$p, q \in V$

4-adj = q in $N_4(p)$

8-adj = q in $N_8(p)$

m-adj = q in $N_4(p)$ OR q in $N_D(p)$ and $N_4(p) \cap N_4(q)$ is null

- 4 → move in -|-
- 8 → move in 4 + Diagonal (may end up in loop)
- m-path Beneficial → No Loop.
  - To check step by step at every point to decide Best path.

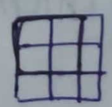
- Distance B/w Pixels - euclidean and manhattan (city Block)
  - criteria (3 cond^n)
    - $D(P,Q) >= 0$ | $D(P,Q) = 0$ if $P = Q$
    - $D(P,Q) = D(Q,P)$ and $D(P,Z) <= D(P,Q) + D(Q,Z)$

Lab arithmetic and logical operation on Images

DIP-5

Ch 3 START

- spatial Transformation → Transformation on set of pixels (neighbour)

 → □ → □ → □ {Stride}

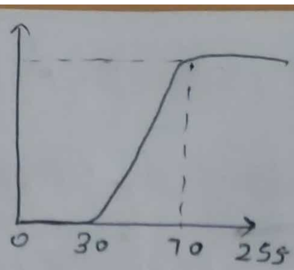
- Intensity Transformation → Transformation - pixel by pixel

[K = no. of colors
  $L = 2^k - 1$]

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

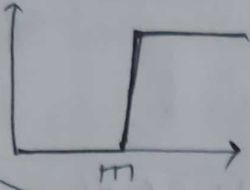
Inverse Transformation → T → $\boxed{L-1-r}$

* Reading Graph →



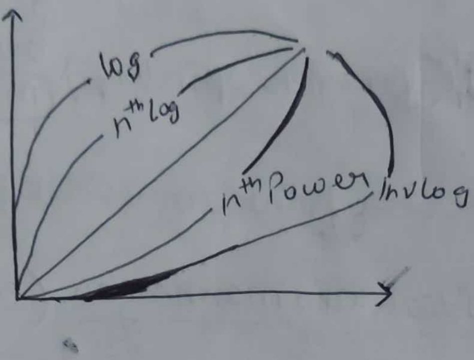
- 0-30 (compressed to 0)
- 30-70
  - Spreading colors
  - highlighting
  - Biasing
- 70-255
  - Replace with 255

Threshold →

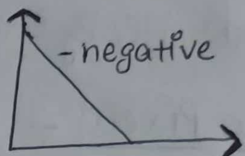


Log < Constrast of Image
small val = lower Cont
- dark to Brigth (not neccessary)
- $c \log(1+r)$

incase → 0



Inv log - undo effect of log.
- Bright to dark (not neccessary)



* Powerlaw → $\boxed{S = Cr^{\gamma}}$ gamma / $S = c(r+\varepsilon)^{\gamma}$

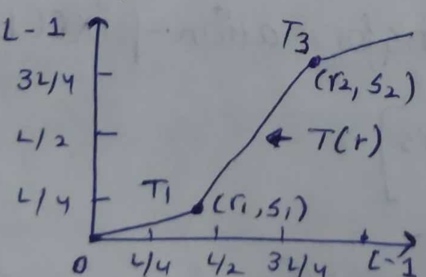
ε → correction value (depends on device)

γ → amount of Constrast and Brigthness

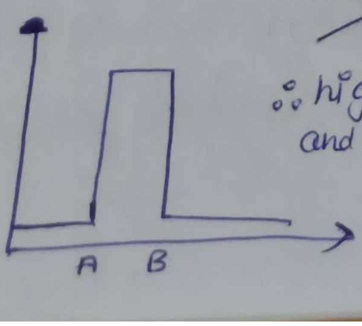
$\gamma < 1 \rightarrow$ log
$\gamma > 1 \rightarrow$ Invlog

* Piecewise Transformation → part to part (particular piece/set Transformation)

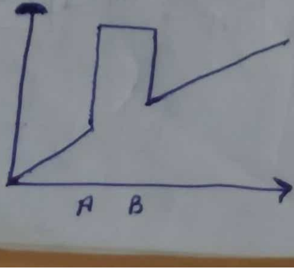
* Constrast stretching → Expands the Range of intensity levels in an Image



$T_1 + T_3$ → Linear/no change



∴ highligths A to B and rest → dark



∴ highlights A to B and rest → No change

Bit- Plane slicing

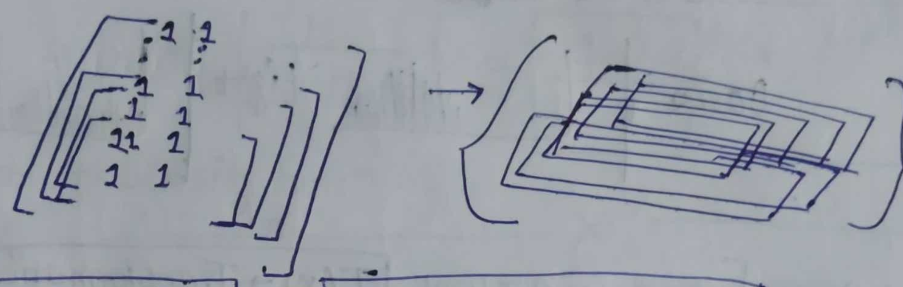
* Represting 255 → Binary → storing every Bit in a Diff Plane /matrix

$K = 8$
$L = 2^8 - 1 = 255$

$L = 2^K$

$$\begin{bmatrix} 255 & 255 & 255 \\ \vdots & \vdots & \vdots \\ 255 & 255 & 255 \end{bmatrix}$$

→ 255 → 1111111 ∵ storing each 1s in a Diff Plane/matrix



K-Bit Plane

- Edges - Boundaries - Corners - Regions = MSB
- Background = LSB

Bit Plane slicing ⇒ slicing out Plane which is not Required

imge ⇄ [ ]

[ ]
[ ]
[ ]
⋮
[ ]

Bit Planes

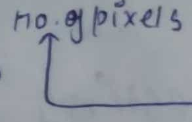
compression → leaving one Bit out (ignoring LSB Bit)

$512 \times 512 \times 8 \rightarrow 512 \times 512 \times 7$

Histogram

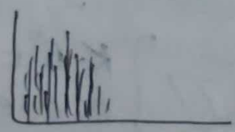
$\boxed{r_K}$ = Intensities of an $\boxed{L\text{-}Level}$ digital Image $f(x,y)$ $[K = 0, 1, 2 \ldots L-1]$

unnormalized histogram of $f \rightarrow h(r_K) = n_K$

$n_K \rightarrow$ no. of pixels in f with Intensity r_K { histogram $\rightarrow$  }

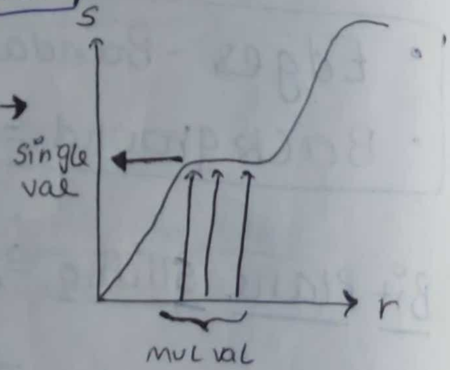
normalized histograms of $f \rightarrow p(r_K) = \frac{h(r_K)}{MN} = \boxed{\frac{n_K}{MN}}$

- Sum of $p(r_K)$ for all value of $K = 1$
- Components of $p(r_K)$ = estimates of the probabilities of Intensity levels occuring in an Image

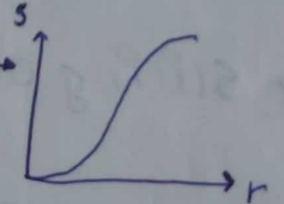
*  Dark |  Light |  High Constrast $\downarrow$ High Diff B/w dark-white Regions

HISTOGRAM EQUALIZATION $\boxed{T(r) \rightarrow \text{Transformation}}$

$\boxed{S = T(r)}$ $\rightarrow$ Monotonic increasing func^n $\rightarrow$



Strictly monotonic Increasing func^n $\rightarrow$



$\boxed{\text{Cond}^n \text{ for HE}}$
- $T(r)/s$ is a monotonic increasing func^n $[0 \leq r \leq L-1]$
- $0 \leq T(r) \leq L-1$ FOR $0 \leq r \leq L-1$

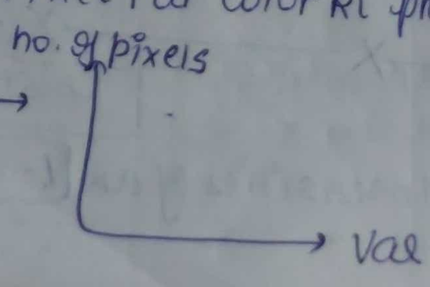
[$\therefore$ Inv is possible on strictly monotonic Increasing func^n]

$\rightarrow \boxed{r = T^{-1}(s)}$ $\rightarrow$ $T(r)$ is a strictly monotonic Increasing Func^n

———— X ———— X ———— X ————

$\boxed{\text{PDF}}$ $\rightarrow$ Probability Distribution func^n { Pixel Pai color ki probab }

- graph = Historgram (Y = no. of pixels) $\rightarrow$



① $p_s(s) = p_r(r) = \left[\frac{dr}{ds}\right]$

② $S = T(r) = L-1 \int_0^r p_r(w)\,dw$ [w = dummy variable]

③ $\frac{ds}{dr} = \frac{dT(r)}{dr}$ ↳ $\frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$

↳ $p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \rightarrow p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right| = \boxed{\frac{1}{L-1}}$

↳ uniform probability Density funcn

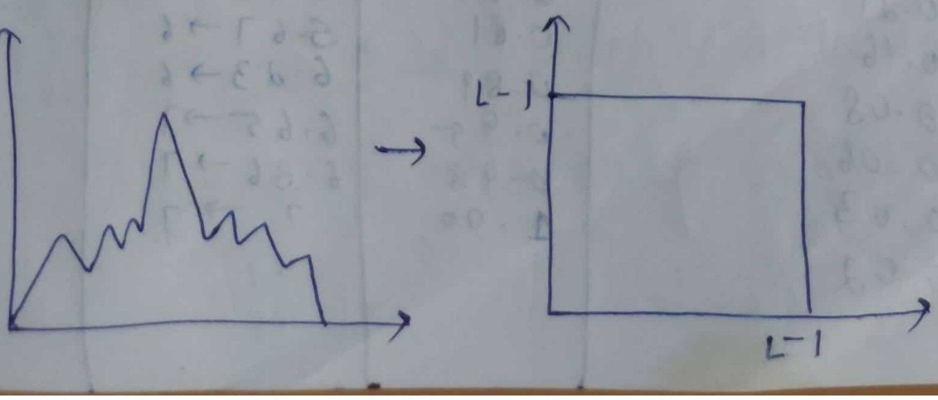
$p_r(r) = \begin{cases} \frac{2r}{(L-1)^2} & 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$ ∴ Continous Value

$S = T(r) = L-1 \int_0^r p_r(w)\,dw + \frac{(L-1)2}{(L-1)^2}\int_0^r w\,dw + \left[\frac{w^2}{2}\right]_0^r \frac{2}{L-1}$

$\rightarrow \frac{r^2}{L-1} \rightarrow \boxed{\frac{r^2}{L-1}} = S$

$\cdot\ \frac{ds}{dr} = \frac{2r}{L-1} \rightarrow \frac{dr}{ds} = \frac{L-1}{2r}$ | $p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| \rightarrow \frac{2r}{(L-1)^2} \times \left(\frac{L-1}{2r}\right) = \boxed{\frac{1}{L-1}}$

∴ Continous values → $\boxed{\int}$ + $\boxed{\text{UDF}}$



$\left\{ \{\int\} + \text{UDF} \rightarrow p_s(s) = \frac{1}{L-1} \right\}$

∴ **Discrete Values**

└ $\boxed{\text{probabilities} + \Sigma}$

$$p_r(r_k) = \frac{n_k}{nm}$$

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

∴ $K = 3 \rightarrow L = 8$] ↳ $7 \sum_{k=0}^{k} p_r(r_j)$

$s_0 = T(r_0) = (L-1) \sum_{j=0}^{0} p_r(r_j) = 7\, p_r(r_0)$

$s_1 = T(r_1) = (L-1) \sum_{j=0}^{1} p_r(r_1) = 7\,[p_r(r_0) + p_r(r_1)]$

$s_2 = 7\,[p_r(r_0) + p_r(r_1) + p_r(r_2)]$

$s_3 = 7\,[p_r(r_0) + p_r(r_1) + p_r(r_2) + p_r(r_3)]$

$s_4 = 7\,[p_r(r_0) + p_r(r_1) + p_r(r_2) + p_r(r_3)]$

⋮

$\boxed{\text{Till } s_7}$ → Round of values of $s_1, s_2, s_3 \ldots s_7$

$\boxed{\text{Digital Histogram Equalisation}}$

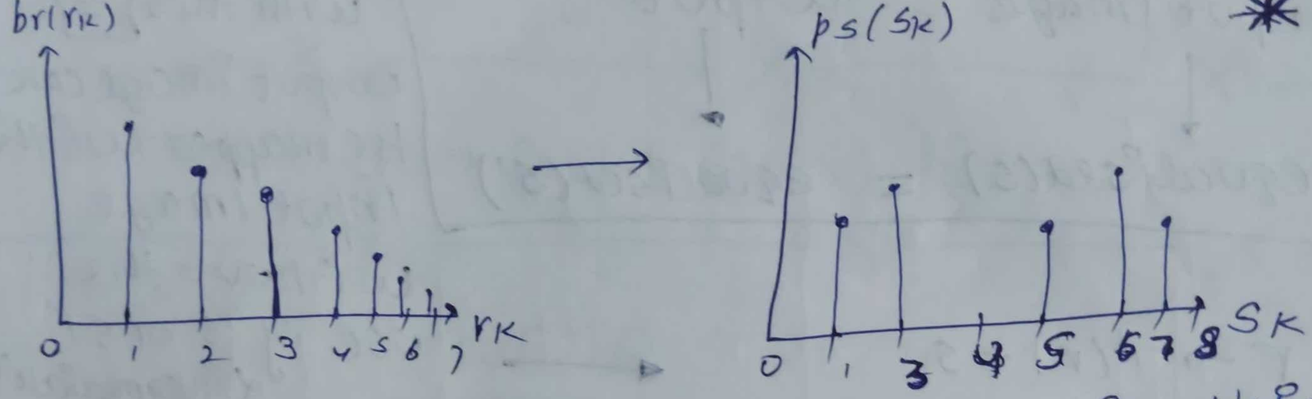
r_k	n_k	$p_r(r_k) = n_k/nm$	$\Sigma p_r(r_k)$	$7\,\Sigma p_r(r_k)$
r_0 0	790	0.19	0.19	1.33 → 1
r_1 1	1023	0.25	0.44	3.08 → 3
r_2 2	850	0.21	0.65	4.55 → 5
r_3 3	656	0.16	0.81	5.67 → 6
r_4 4	329	0.08	0.89	6.23 → 6
r_5 5	245	0.06	0.95	6.65 → 7
r_6 6	122	0.03	0.98	6.86 → 7
r_7	81	0.02	1.00	7 → 7

Result → 790 pixels equalised to 1

1023 ⟶ 3
850 ⟶ 5
(656+329) = 985 → 6
(245+122+81) = 448 → 7

Equalised

Histogram shape is not retained



→ No New Intensity levels are allowed in this Method (Σ → discrete)

→ FLAT histograms are rare in this process

* Continous / Results in Uniform histograms But Discrete may or may not

---

Inverse log T → Undo the effect of log Transformation
Applied to low-Constrast Image to enchance their constrast

Log T → enchance the Constrast of Image
- high c → High Constrast
- low c → low Constrast

Histogram Matching

- Matching specefically
- Redistributing the pixels according to output.

→ Input Image → equalised(s) = output → equalised(s')

with this [=] output image can be mapped with Input Image without the use of s or s' (intermediate)

→ $r \to T(r) = s$

→ $z \to G(z) = s$

$z = G^{-1}(s)$

$z = G^{-1}[T(r)]$

IMP
- changes INPUT IMAGE.
- Transforming the intensities of the pixel in INPUT IMAGE so that the output Image is = desired

- $p_r(r) = \frac{2r}{(L-1)^2}$ | • $p_z(z) = \frac{3z^2}{(L-1)^3}$ | $s = (L-1)\int_0^r p_r(w)\,dw$

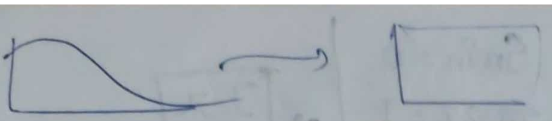
$s = (L-1)\int_0^r \frac{2w}{(L-1)^2}\,dw \to \frac{2(L-1)}{(L-1)^2}\int_0^r w\,dw$

$\to \frac{\cancel{2}}{L-1}\left[\frac{w^2}{\cancel{2}}\right]_0^r \to \left[\frac{w^2}{L-1}\right]_0^r = \boxed{\frac{r^2}{L-1}}$

x ——— x ——— x ——— x

$\to s' = (L-1)\int_0^z p_z(t)\,dt \to (L-1)\int_0^z \frac{3t^2}{(L-1)^3} \to \frac{3}{(L-1)^2}\int_0^z t^2\,dt$

$\to \frac{\cancel{3}}{(L-1)^2}\left[\frac{t^3}{\cancel{3}}\right]_0^z \to \boxed{\frac{z^3}{(L-1)^2}}$



→ $\boxed{s = s'}$ → $\frac{r^2}{\cancel{L-1}} = \frac{z^3}{\cancel{(L-1)^2}}$ $(L-1)$ = $\boxed{z = \left((L-1)r^2\right)^{1/3}}$ rel^n B/W z and r

---X---X---X---X---

Mapping Towards → High Value (monotonic Increasing)

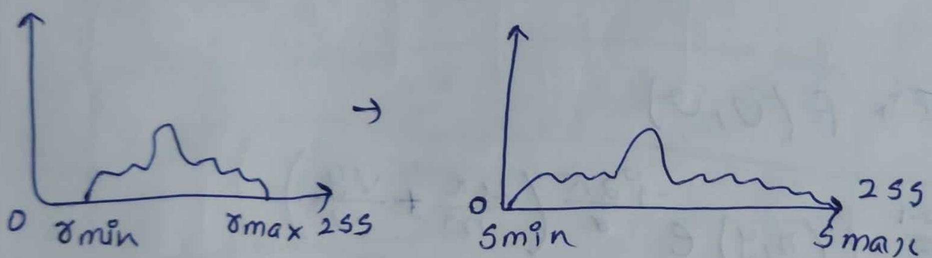
---X---X---X---

Also known as → Double (2 time) Histogram Equalization

---X---X---X---

LINEAR STRECHING

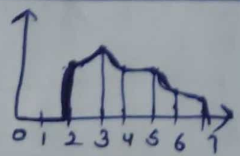
- Streching End points → Histogram retains shape



$$s = T(r) = \underbrace{\frac{(s_{max} - s_{min})}{(r_{max} - r_{min})} (r - r_{min})}_{\text{scaling}} + \underbrace{s_{min}}_{\text{Translating}}$$

$(a \oplus b)$ — Maintaining proportion ↓ Maintaing shape.

r	
→ 0	0
1	0
2	50 — rmin
3	60
4	50
5	20
6	10 — rmax
7	0

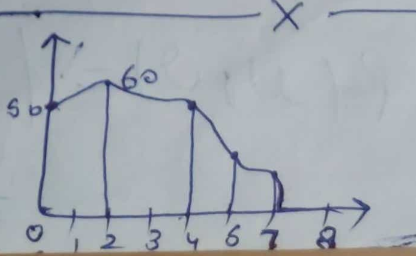


$s_{min} = 0 \mid s_{max} = 7$

$s = T(r) = \frac{7}{4}(x-2) = \frac{7x-14}{4}$

---X---X---

r =	map	s =
2	←	0
3		2
4		4
5		5
6		7



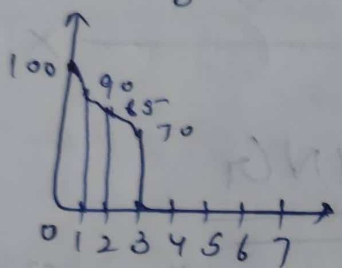
Question →

0	100 — r_{min}
1	90
2	85
3	70 → r_{max}
4	0
5	0
6	0
7	0

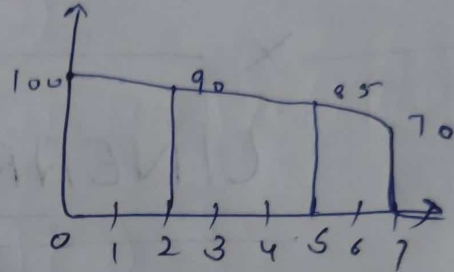
$S_{min} = 0$
$S_{max} = 7$ → $\boxed{\frac{7}{3}x}$

r =	= S
0	0
1	2
2	5
3	7

r



S



* 2 2D DFT

→ $f(x,y) \xleftrightarrow{2D\,DFT} F(u,v)$

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

- $|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$
- $\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
- $P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$.

Fourier Transform

→ Img enchancement in freq Domain

1 1D Discrete Fourier Transform (1D DFT)

$\underbrace{x(n)}_{\text{Spatial Domain}} \xleftrightarrow{DFT} \underbrace{X(K)}_{\text{freq Domain}}$ | N → No. of samples

$$\boxed{X(K) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}Kn}}$$

$\underbrace{X(K)}_{\text{freq Domain}} \xleftrightarrow{IDFT} \underbrace{x(n)}_{\text{Spatial Domain}}$

$$\boxed{x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}Kn}}$$

* Fourier spectrum

$$\boxed{|F(U)| = \left[R^2(U) + I^2(U)\right]^{1/2}}$$

* Phase Angle

$$\boxed{\phi(U) = \tan^{-1}\left|\frac{I(U)}{R(U)}\right|}$$

* Power spectrum

$$\boxed{P(U) = |F(U)|^2 = R^2(U) + I^2(U)}$$

* filtering in freq Domain

① Image $f(x,y)$ of size $M \times N$

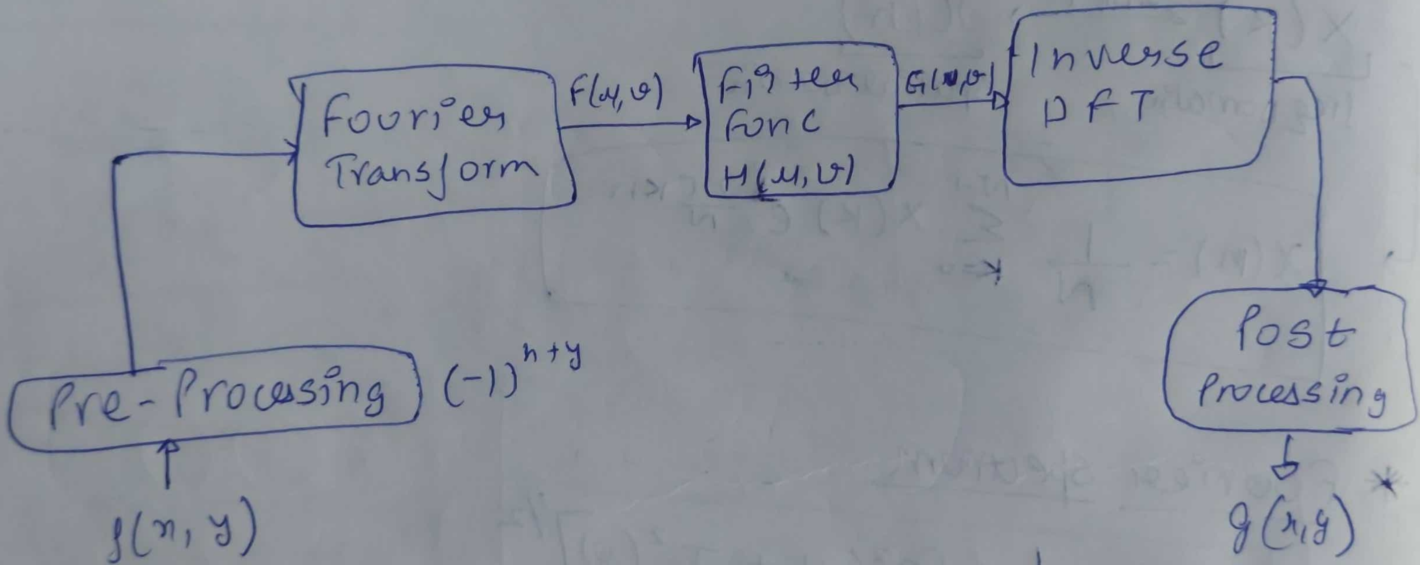
② Pre-process → $f(x,y)\,(-1)^{x+y}$

③ Fourier Transform → $F(u,v)$ = F.T of $f(x,y)$

④ filter function → $H(u,v)$

↳ $\boxed{G(u,v) = H(u,v) \cdot F(u,v)}$

⑤ $g(x,y) = \text{IDFT}\,[G(u,v)]\,(-1)^{x+y}$.



$\boxed{\text{Pre-processing}}$ → center the Image

↳ center of Spatial Domain (top left corner) → center of freq Domain

→ low freq around the center and high freq near the edges of Spectrum

---

$\boxed{\text{Post Processing}}$ → Undo effects of Pre-processing

(a) Low Pass FILTERS

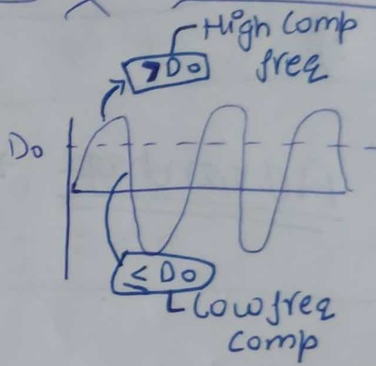
- Removes High freq comp & Retains low freq comp
- Smooth the Image

(b) High Pass FILTERS

- Removes low freq comp & Retains High freq comp
- Sharping the Image

---

① IDEAL LOW PASS FILTER



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

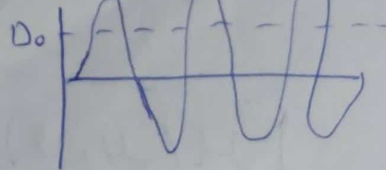
D_0 = non negative quantity
$D(u,v)$ = Distance from Point (u,v)

$\boxed{f(x,y) \rightarrow M \times N}$ $\quad D(u,v) = \left[u - \frac{M}{2}\right]^2 + \left[v - \frac{N}{2}\right]^2$

---

② IDEAL HIGH PASS FILTERS

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



DISADVANTAGE → BLURRED EDGES

BUTTERWORTH LPF

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

BUTTERWORTH HPF

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

Advantage → useful in defining the edges.

GAUSSIAN LPF

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

:→ Removes low freq Noises

GAUSSIAN HPF

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

:→ Removes High freq noises

Image Restoration → Removing Noise from Image

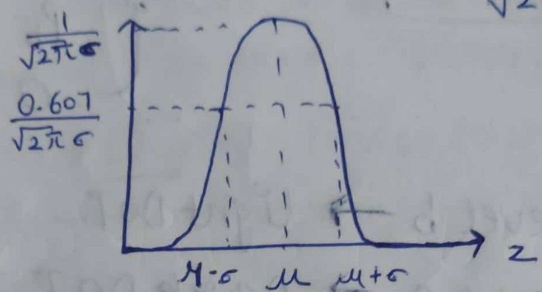
$$g(x,y) = f(x,y) + \eta(x,y)$$

* **Salt & Pepper Noise** — Randomly scattered black + white pixels
  └ Impulse

---

* **Gaussian Noise** → PDF = $P(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}$

  z = gray level
  μ = μ of z



---

* **Rayleigh Noise** → PDF = $\begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & ; z \geq a \\ 0 & ; z < a \end{cases}$

mean: $\mu = a + \sqrt{\pi b/4}$

Variance: $\sigma^2 = \frac{b(4-\pi)}{4}$ ←

---

* **Erlang [Gamma] Noise** → $P(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & ; z \geq 0 \\ 0 & ; z < 0 \end{cases}$

mean: $\mu = \frac{b}{a}$

Variance: $\sigma^2 = \frac{b}{a^2}$

$\boxed{b=0}$

---

* **Exponential noise** → $P(z) = \begin{cases} ae^{-az} & ; z \geq 0 \\ 0 & ; z < 0 \end{cases}$

$\boxed{\mu = \frac{1}{a}}$ $\boxed{\sigma^2 = \frac{1}{a^2}}$

* **Uniform noise** → $P(z) = \begin{cases} \frac{1}{b-a} ; & a \le z \le b \\ 0 ; & \text{otherwise} \end{cases}$

$\mu = \frac{a+b}{2}$

$\sigma^2 = \frac{(a+b)^2}{12}$

---

* **Salt & Pepper Noise** → $P(z) = \begin{cases} P_a ; & z = a \\ P_b ; & z = b \\ 0 ; & \text{otherwise} \end{cases}$

1) $b > a \Rightarrow$ gray level b → Light DOT
gray level a → Dark DOT

Periodic Noises

1) while Image Acquisition → electromechanical Interference

2) Inspection of Fourier Spectrum

3) If Image strips are available → use data from strips (subImage(s)) →
calculating → mean & variance of gray level

$\mu = \sum_{z_i \in S} z_i P(z_i)$ → gray level values

$\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 P(z_i)$ – normalized Histogram

SPATIAL FILTERING TO REMOVE NOISE

$g(x,y) = f(x,y) + \eta(x,y) \rightarrow \hat{f}(x,y)$

I. Mean filters

(i) Arithmetic mean

$$\boxed{\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)}$$

$S_{xy} \rightarrow$ Set of coordinates in a rectangular subimage window of size $m \times n$. Center at point (x,y)

→ smoothing filter → Blur the Image to remove noise

(ii) Geometric mean

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

[Π = product]

→ Smoothing similar to arithmetic mean filter

→ Tends to lose Image details

(iii) Harmonic Mean

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

→ works well for Salt Noise and Gaussian Noise

→ Fails for Pepper Noise

(iv) Contra Harmonic Mean

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

$\boxed{Q \rightarrow \text{order of FILTER}}$

Reduce effect of salt & pepper noise

$Q \rightarrow +ve \rightarrow$ eliminates Pepper noise
$Q \rightarrow -ve \rightarrow$ eliminates salt noise

$Q = 1 \rightarrow$ Arithematic Mean

II. ORDER STATISTICS FILTER

(i) MEDIAN FILTER

$$\boxed{\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\}}$$

DISADVANTAGE → works well when impulse noise is not large

- → Replace the value of pixel by the median of gray levels
- → Excellent noise Reduction with less Blurring
- → Effective in presence of {→ Biploar, → unipolar, → Impulse} noise

---

(ii) min & max Filters

$$\boxed{\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{max}} \{g(s,t)\}}$$ → ↓ Pepper Noise

→ FIND the Darkest Point

$$\boxed{\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}}$$ — ↓ salt Noise

→ FIND the Brightest Point

(iii) Midpoint FILTER

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\}\right]$$

→ WORKS BEST → Randomly Distributed Noise (Gaussian & uniform)

(iv) Alpha-trimmed mean FILTER

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

we get

$\boxed{mn-d}$ → Subtracting $\frac{d}{2}$ lowest & $\frac{d}{2}$ highest from gray level $\left[\frac{d}{2}+\frac{d}{2}=d\right]$

$\boxed{d=0}$ → Arithmetic mean filter

$\boxed{d = \frac{mn-1}{2}}$ → median filter

ADAPTIVE FILTERS

① → Adaptive local noise Reduction

- Mean → measure of avg gray level values
- Variance → measure of avg constrast
- Local region operation → S_{xy}

FILTER BASED on 4 Quantities

ⓐ $g(x,y)$ = value of noisy Image at (x,y)

ⓑ σ_n^2 = variance of noise corropting $f(x,y)$ to form $g(x,y)$

ⓒ m_L → Local mean of the pixels in S_{xy}

ⓓ σ_L^2 → Local variance of the pixels in S_{xy}

$$\boxed{\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2}\left[g(x,y) - m_L\right]}$$

② Adaptive median filter

→ Can handle large values of impulse noise.

→ Sol^n to Disdavantage of median filter

→ Preserves details of Image while Smoothing non-Impulse noise.

Z_{min} = min value of gray level in S_{xy}

Z_{max} = max valuee . .

Z_{med} = median values . .

S_{max} = max allocated size of S_{xy}

Works in 2 levels

Level A

$A_1 = Z_{med} - Z_{min}$

$A_2 = Z_{med} - Z_{max}$

If $\boxed{A_1 > 0}$ & $\boxed{A_2 < 0}$ GOTO Level B

ElSE → Increase the window size

if window $\leq S_{max}$ Repeat level A

else O/p Z_{xy}

0) Initial A_1 & A_2
1) If $A_1 > 0$ & $A_2 < 0$ goto 5
2) window size ++
3) if window size $\leq S_{max}$ goto 0
4) O/p Z_{xy}

Level B

5) $B_1 = Z_{xy} - Z_{min}$
6) $B_2 = Z_{xy} - Z_{max}$
7) If $B1 > 0$ & $B_2 < 0$ goto 9
8) O/p Z_{med}
9) O/p Z_{xy}

---

① Remove salt & Pepper Noise | ② Provide smoothing.

PERIODIC NOISE REDUCTION USING FREQ DOMAIN FILTERING

① Band Reject filters

→ ~~Attenuate~~ a Band of frequencies above the origin of the F.T

Attenuate = Removes

Ideal

$$H(u,v) = \begin{cases} 1 \; ; & D(u,v) < D_0 - \frac{w}{2} \\ 0 \; ; & D_0 - \frac{w}{2} \le D(u,v) \le D_0 + \frac{w}{2} \\ 1 \; ; & D(u,v) > D_0 + \frac{w}{2} \end{cases}$$

Remove

$D(u,v)$ → distance from the origin of the centered freq band

w → width of Band

D_0 → radial center / Cutoff freq

Butterworth

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

(n → order of filter)

Gaussian

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)\, w}\right]}$$

② Band Pass FILTERS

→ Opposite Operation of Band reject

→ will allow the Band of freq above the origin F.T Pass the ↑

$$\boxed{H_{bp}(u,v) = 1 - H_{br}(u,v)} \quad *$$

Ideal → 0 / 1 / 0	BW → $1 - \dfrac{1}{1+\left[\dfrac{D(u,v)\,W}{D^2(u,v) - D_0^2}\right]^{2n}}$	G → $e^{-\frac{1}{2}\left[\frac{D^2(u,v) - D_0^2}{D(u,v)\,W}\right]}$

— X — X —

(iii) NOTCH FILTERS

H_{np} — notch Pass

H_{nr} — notch Reject

→ passes freq in Pre-defined in neighborhoods

Ideal Radius D_0 | center (u_0, v_0) |

$$H(u,v) = \begin{cases} 0; & D_1(u,v) \le D_0 \text{ OR } D_2(u,v) \le D_0 \\ 1; & \text{otherwise} \end{cases}$$

$$D_1(u,v) = \left[\left(u - \frac{M}{2 - u_0}\right)^2 + \left(v - \frac{N}{2 - v_0}\right)^2\right]^{1/2}$$

$$D_2(u,v) = \left[\left(u - \frac{M}{2 + u_0}\right)^2 + \left(v - \frac{N}{2 + v_0}\right)^2\right]^{1/2}$$

Butterworth

$$H(u,v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u,v)\,D_2(u,v)}\right]^n}$$

— X —

Gaussian

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D_1(u,v)\,D_2(u,v)}{D_0^2}\right]}$$

— X — X —

HIGH PASS FILTERS $(u_0 = v_0 = 0)$

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

R → **data Redundancy**

$R = \boxed{1 - \frac{1}{C}}$

C → **Compression Ratio**

$C = b/b'$

Encoder → Compression

Decoder → Decompression

Codec → Device

$f(x,y)$ → encoder → Decoder → output Image $f'(x,y)$

Encoding → used to remove redundancies through a series of 3 independent operation

- a) **mapper** → reduce spatial & temporal redundancies
- b) **Quantizer** → keeps Irrelevant Information out of compressed representation
- c) **Symbol Encoder** → Generates a var length code to rep the quantizer output

---

$f(x,y)$ → Mapper → Quantizer → Symbol Encoder

→ Symbol Decoder → Inverse mapper → $f'(x,y)$

Q1) RLC on Run-length coding

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

Horizontal line scan

→ [0, 2] [1, 5]
[0, 6] [1, 1]
[0, 3] [1, 4]
[0, 4] [1, 3]
[0, 1] [1, 6]

→ max length → $\boxed{6}$

→ 6 → 110 → $\boxed{3 \text{ BITS}}$

→ TOTAL VECTORS → 10

→ No. of BITS PER PIXEL = 1 ['0' or '1']

Ans → Total no. of pixels → $10 \times (3+1) = 40$ bits
→ b'

b → Total pixel bits in org image = $7 \times 5 = 35$

→ $C = \frac{b}{b'} = \frac{35}{40} = \frac{7}{8} = 0.87$

Vertical Line Scan

→ [0,2] [1, 3]
[0,3] [1, 2]
[0, 1] [1, 4]
[0, 1] [1, 4]
[0,3] [1, 2]
[0,2] [1, 3]
[0, 4] [1, 1]

→ max length → 4 → 100 → $\boxed{\text{3 BITS}}$

→ Total vectors → 14

→ No. of BITS PER PIXEL → 1 ['0' or '1']

Ans → $14 \times (3+1) = 56$ bits → b'

$b \to 7 \times 5 = 35$

$C \to \frac{b}{b'} \to \frac{35}{56} \Rightarrow 0.625$

$\boxed{\text{Redundancy} = 1 - \eta}$

$\boxed{\text{Entropy } \sum_{i=1}^{n} P_i \log_2 \left(\frac{1}{P_i}\right)}$

$\boxed{\eta = \frac{H}{L_i}}$ – Efficiency

Q 2) Huffman Coding

→

A	B	C	D
0.4	0.3	0.2	0.1

→

A 0.4 (1) → 0.4 (1) → 0.6 0
B 0.3 (00) → 0.3 (00) → 0.4 1
C 0.2 (010) → 0.3 (10)
D 0.1 (110)

$\boxed{L_i = \text{length}}$

A	B	C	D
1	00	010	110

length of code

$L_i = \sum_{i=1}^{n} P_i (L)$

$\to 0.4(1) + 0.3(2) + 0.2(3) + 0.1(3)$

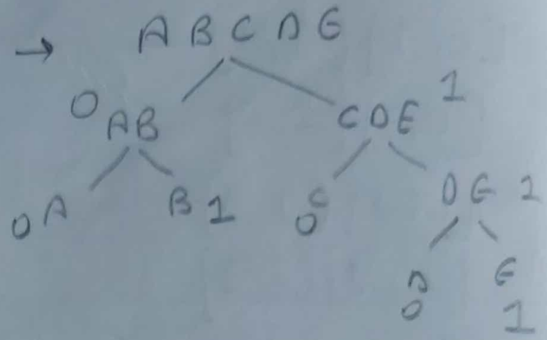
Q3 Shanon fanon

→

A	B	C	D	E
12	8	7	6	5

↳

A	12	0	00	
B	8		01	
C	7	1	10	
D	6		11	110
E	5			111

→ 

↳

A	B	C	D	E
00	01	10	110	111

$L \Rightarrow 12(2) + 8(2) + 7(2) + 6(3) + 5(3)$

$H \Rightarrow p_i \log_2\left(\frac{1}{p_i}\right) \rightarrow 12 \log_2\left(\frac{1}{12}\right) + \ldots + 5 \log_2\left(\frac{1}{5}\right)$

$\eta = \frac{H}{L}$

$R = 1 - \eta$

Arithmetic coding

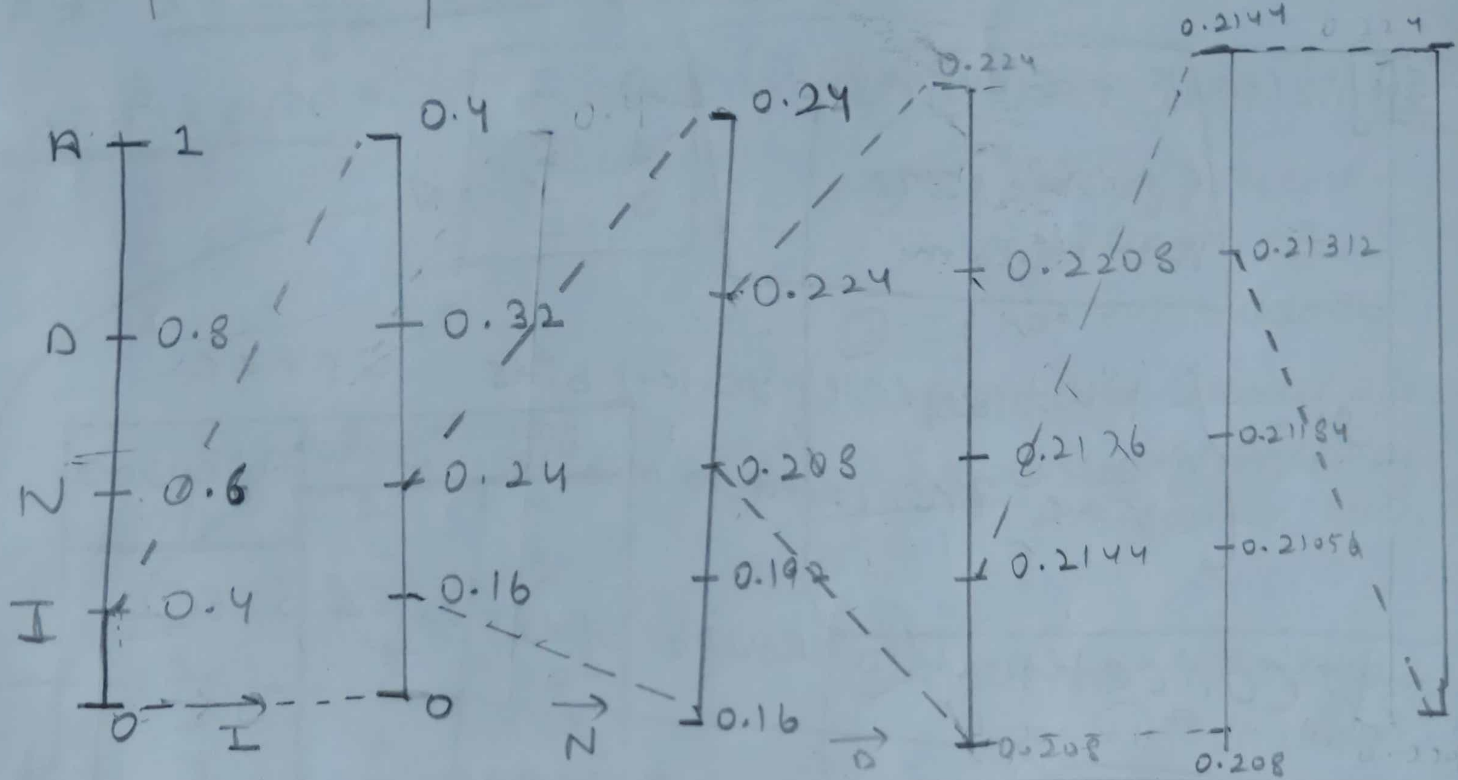
→ make Table → freq from input → Cumulative freq

→ Line Range 0 to 1

I/p → $\boxed{\text{INDIA}}$

	f	cf
I	2/5 = 0.4	0.4
N	1/5 = 0.2	0.6
D	1/5 = 0.2	0.8
A	1/5 = 0.2	1

* START Encoding from the start letter



(I)

$UP_I = LL + (UP - LL)(P(symbol))$

$0 + 0.4(0.4) = 0.16$

$UP_N = LL + (UP - LL)(P(symbol))$

$0 + 0.4(0.6) = 0.24$

(N)

$UP_I \Rightarrow 0.16 + 0.08(0.4) = 0.192$

$UP_N \rightarrow 0.16 + 0.08(0.6) = 0.208$

$UP_D \rightarrow 0.16 + 0.08(0.8) = 0.224$

(D)

$UP_I \rightarrow 0.208 + (0.016)(0.4)$ $\underline{0.2144}$

$UP_N \rightarrow 0.208 + (0.016)(0.6)$ $\underline{0.2176}$

$UP_D \rightarrow 0.208 + (0.016)(0.8)$ $\underline{0.2208}$

(I) $\rightarrow 0.208 + (0.0064)(0.4)$ $\boxed{0.21056}$

$UP_N \rightarrow 0.21184$

$UP_D \rightarrow 0.21312$

Do these steps Acc to the letters of input

Golomb code ($G_m(n)$)

n — integer no.
m — Divisor (m)

S1] unary code of $q = \text{floor}(n/m)$

S2] let
$$\boxed{\begin{aligned} k &= \text{ciel}(\log_2 m) \\ c &= 2^k - m \\ r &= n \% m \end{aligned}}$$

$$r' = \begin{cases} r \text{ truncated to } k-1 \text{ bits} & 0 \le r < c \\ r+c \text{ truncated to } k \text{ bits} & \text{otherwise} \end{cases}$$

S3] concetenate result of STEP 1 & STEP 2

Q → GC for 9 with divisor 4

$n = 9$
$m = 4$] $G_4(9) = ?$

S1] $q = 2 \rightarrow 110$ (no. of q one's + followed by 0)

S2] $k = 2$
$c = 0$
$r = 1$

$r' \rightarrow r + c = 1$ (1 rep in 2 bits)
$r' \rightarrow \underline{01}$

Ans $\boxed{11001}$

LZW Encoding

1) Dictionary Based

1) p → ab ab ab ab ab c ab ab ab ba

Initial dict →

1	a
2	b
3	c

Encoded o/p	Dictionary	
–	1	a
–	2	b
1	3	c
2	4	ab
4	5	ba
5	6	abb
2	7	bab
3	8	bc
4	9	ca
6	10	aba
1	11	abba
	–	–

① START with single

② If already present inc the length +1

③ If not exist add to table and Encoded o/p would be the index of first formed letter

④ After Addition start from (+1) th character

Decoding → b- routhao and ab first rakhdo ba

→ 1 2 4 5 2 3 4 6 1

→ encoding of min 2 character

Decoding

comb^n of letters if not in initial

Recieved	Decode	Consider initial Dict			
		Index	entry	index	Partial Entry
1	a	–	–	4	a-
2	b	4	ab	5	b-
4	ab	5	ba	6	ab-
5	ba	6	abb	7	ba-
2	b	7	bab	8	b-
3	c	8	bc	9	c-
4	ab	9	ca	10	ab-
6	abb	10	aba	11	abb-
1					

Loseless Predictive

① $f(n)$

② $\hat{f}(n) = \text{round}\left[\sum_{i=1}^{n} \alpha_i f(n-i)\right]$

$\quad$ ↳ $n = 1 \mid \alpha = 1$

$\quad$ ↳ $\text{round}[f(n-1)]$

$\quad$ ↳ $\boxed{\hat{f}(n) = f(n-1)}$

③ $\boxed{e(n) = f(n) - \hat{f}(n)}$

④ $\boxed{f(n) = e(n) + \hat{f}(n)}$

Q → $\{23, 34, 39, 47, 55, 63\} \rightarrow O_1$

$f(n)$	$\hat{f}(n)$	$e(n)$
23	0	23
34	23	11
39	34	5
47	39	8
55	47	8
63	55	8

encoder

→ O_2

max no. in O_2 = 63 → 6 bits + 1 bit for sign

So $\underbrace{6}_{\text{no of input}} \times 7 = \underline{\underline{42}}$ → bits for org message max no. in $O_2 \rightarrow 23$

$\downarrow$

5 bits + 1 = 6

$\hookrightarrow \boxed{6 \times 6 = 36}$ bits for encoded message

$\rightarrow$ Saving $42 - 36 = 6$ bits

$e(n)$	$f^{\wedge}(n)$	$f(n)$
23	0	23
11	23	34
5	34	39
8	39	47
8	47	55
8	55	63

Decoder

$\rightarrow O_3$

$\boxed{O_1 = O_3 = \text{No loss}}$

(a) DILATION

$\hat{B} = \{w \mid w \in -b,\ b \in B\}$

$(A)_z = \{c \mid c \in a+z,\ a \in A\}$

① Process of expanding Image

② Increases Brightness of Image

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\}$$

en

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

Image

$0 \neq 1$

structural Image/mask

origin

L

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

Rep with SE

→ Conv SE/SI/mask to Image & replace only when origin value = image value at pos

(S1)
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(S2)
$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(S3)
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Increment

(S4)
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(S5)
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Errosion

→ Image shrinking

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

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

Image

origin

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

mask

FIND AREA similar to mask & replace the origin position with 1 and rest to zero

→
$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$\boxed{3 \times 3 = \text{center origin}}$

OPENING

- Erossion followed by Dilation
- $\boxed{A \circ B = (A \ominus B) \oplus B}$
- Identify gaps in an Image

CLOSING

- DILATION followed by Erossion
- $\boxed{A \bullet B = (A \oplus B) \ominus B}$
- eliminate small holes

Duality $\rightarrow (A \bullet B)^c = (A^c \circ \hat{B})$

Boundary Extraction $\rightarrow A - (A \ominus B)$

Region filling $\rightarrow \boxed{X_k = (X_{k-1} \oplus B) \cap A^c}$

$\boxed{X_0 = A = Img}$ start Point

$\boxed{X_1 = (X_0 + B) \cap A^c}$

$\boxed{X_2 = (X_1 + B) \cap A^c}$

Connected Components

$$\boxed{X_k = (X_{k-1} \oplus B) \cap A}$$

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

x_0 = any pixel with value 1

HIT OR MISS → FINDING SHAPES

$$\boxed{A \circledast B = (A \ominus X) \cap [A^c \ominus (W - X)]}$$

Image Segmentation

* → Point detection

→ 2nd order laplacian filter → $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

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

Input Image → Padding → Padded image × filter

$\downarrow z$

$\boxed{z}$

Line detection

→ Thick lines → 1st order derivatives

→ Thin lines → 2nd order derivatives

$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$ Horizontal

$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$ Vertical

$+45°$ $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

$-45°$ $\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$

Edge detection

→ Thicker → first order → Sobel & Robert cross

Thinner → 2nd order → Laplacian

Laplace

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

RC →

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

Sobel

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

LoG → Laplacian of Gaussian

$$\boxed{\nabla^2 f(x,y) = \frac{d^2 f}{dx^2} + \frac{d^2 f}{dy^2}}$$

$$\nabla^2 \left(G(x,y) * f(x,y) \right) = \nabla^2 G(x,y) * f(x,y)$$

$$\nabla^2 G(x,y) = \left[\frac{d^2}{dx^2} + \frac{d^2}{dy^2} \right] G(x,y)$$

$$= \left[\frac{d^2}{dx^2} + \frac{d^2}{dy^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(\frac{x^2+y^2-2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussian blurs the image → ↓ intensity of structures at scales smaller than σ

DoG Differences of Gaussian

$$\rightarrow \left(\frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}}\right) - \left(\frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}\right) \quad \boxed{\sigma_1 > \sigma_2}$$

Thresholding

$$\boxed{g(x,y) = \begin{cases} 1 & \text{if } (f(x,y)) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}}$$

Global: T is constant

Variable: T changes over time

Local or Regional → In variable if T at any point (x,y) depends on the properties of a neighborhood of (x,y)

Dynamic → In variable T, if the value of T depends on the spatial coordinates (x,y).