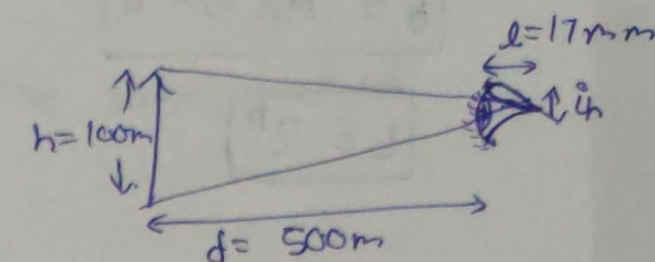
Lecture 3 06/07/24

Image formation in Eye.

Here, i_h is the height of image formed in the eye.



Q) $h = 100m$, $d = 500m$, $l = 17mm$,
$i_h = ?$

Soln.

$$\frac{i_h}{l} = \frac{h}{d}$$

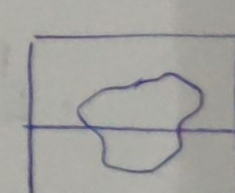
$$\Rightarrow i_h = \frac{h}{d} \times l = \frac{100}{500} \times 17mm = 3.4mm.$$

Brightness adaptation and ~~des~~discrimination of human eye.

- Light is a form of energy.

$$\text{Intensity } f^n = f(x,y) = \underbrace{r(x,y)}_{\text{Reflection co-efficient}} * \overbrace{i(x,y)}^{\text{illumination component}}$$

Take continuous image → Scan a line → Sample it → Quantize it.



Intensity Matrix

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

Each element is known as → image element, pel, pixel.
Their values → intensity or gray level.

Here, x and y are spatial (or plane) coordinates.

Number of bits (b) required to store digitized image,

$\boxed{b = M \times N \times k}$

$\boxed{L = 2^k}$

M → pixels in a column (No. of rows)

N → pixels in a row (No. of column)

k → Constant

L → No. of intensity levels.

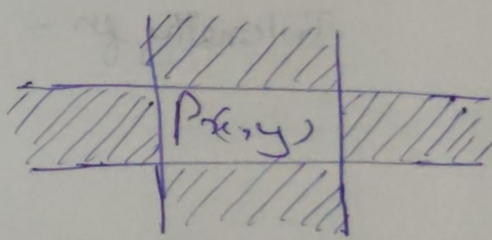
Ex: if $k = 3$
$L = 8$ levels of intensity

if $k = 1$
$L = 2^1 = 2$
Levels → 0 and 1

Neighbors of Pixel

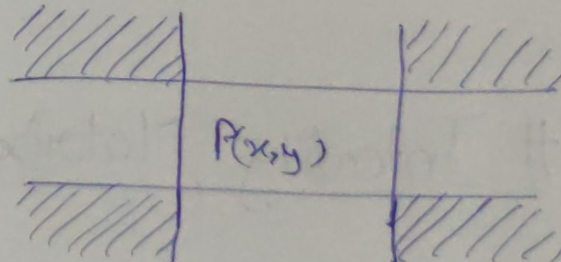
08/08/24

- Four-Neighbours of P(x,y)



$N_4(P)$ ⇒ Neighbours are
$(x+1, y)$, $(x-1, y)$,
$(x, y+1)$, $(x, y-1)$

- ~~Eight~~ Diagonal - Neighbours of P(x,y)



$N_D(P)$ ⇒ $(x-1, y-1)$,
$(x+1, y+1)$,
$(x-1, y+1)$,
$(x+1, y-1)$

- Eight - Neighbours of P(x,y)

$N_8(P) = N_4(P)$ and $N_D(P)$

Adjacency

→ Relation between neighbour values. of two pels. (let p and q)

- 4-adjacency → Relation betn the same valued 4-neighbours.
- 8-adjacency → —"— 8-neighbours.

Note: Prefer diagonal connections for 8-adjacency.

- m-adjacency →
  i) q is in $N_4(P)$

  OR, ii) q is in $N_D(P)$ and set $N_4(P) \cap N_4(q)$ has no pixel values of set V.

Note: Prefer 4-N path, while doing m-adjacency.

Paths

- 4-Path → Form path traversing through the given set of values V, via 4 Neighbours.
- 8-Path → —"— 8 Neighbours.

Note: Prefer the diagonal path.

- m-~~adjacency~~ Path →

Preference
(i) $N_4(P)$
(ii) $N_D(P)$
and $N_4(P) \cap N_4(Q) \neq$ any value in set V.

Note: Don't forget to check the intersection for Diagonal ~~path~~ Neighbour path.

(Abs of ...)

..of ...)

..ds .

Q) Consider the two image subsets S_1 and S_2 shown in the following figure. For $V = \{1\}$ determine whether these two subsets are –

a) 4 – adjacent
b) 8 – adjacent
c) m – adjacent

	S_1				S_2				
0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1 (Q)	1	0	0	0
0	0	1	1	1 (P)	0	0	0	0	0
	0	1	1	1	0	0	1	1	

→ $N_4(P) \cap N_4(Q)$

Soln a) 4 – adjacency :
$\because$ Q is not in the set $N_4(P)$
$\therefore$ S_1 & S_2 are not 4-adjacent.

b) 8 – adjacency:
$\because$ Q is in the set $N_8(P)$
$\therefore$ S_1 & S_2 are 8 – adjacent.

c) m – adjacency
$\because$ Q is in $N_D(P)$ and $N_4(P) \cap N_4(Q)$ has no pixel value of V.
$\therefore$ S_1 & S_2 are m – adjacent.

Q-2) a) Let $V = \{0, 1\}$ and compute the lengths of shortest 4, 8 & m-path b/w P & Q. If a particular path doesn't exist between these two points explain why?

b) Repeat for $V = \{1, 2\}$

	3	1	2	1 (Q)
	2	2	0	2
	1	2	1	1
(P)	1	0	1	2

Sol^n(a) 4-path

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

$\because V = \{0, 1\}$

$\therefore$ We have to traverse through 0 and 1.

It is impossible to make a path betn P & Q using 4-adjacency, having the values of set V.

8-path

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

It is possible to make a path betn P & Q using 8-adjacency, having the values of set V.

$\Rightarrow$ length = 5.

m-path

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

It is possible to make a path betn P & Q using m-adjacency having the values of set V.

$\Rightarrow$ length = 5

Sol^n(b) Now $V = \{1, 2\}$.

4-path

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

Path is possible.

length = 6

8-path

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

Path is possible

length = 4.

m-path

Path is possible

length = 6

$$\begin{matrix} 3 & 1 & 2 & 1 \; (Q) \\ 2 & 2 & 0 & 2 \\ 1 & 2 & 1 & 1 \\ (P)\; 1 & 0 & 1 & 2 \end{matrix}$$

12/08/24

Distance Measures

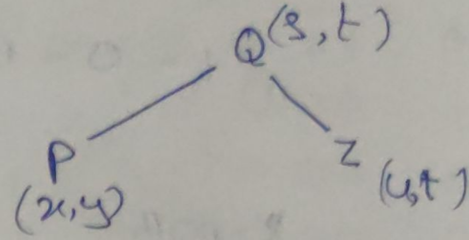
Given pixels p, q and z with coordinates (x, y), (s, t) and (u, t).

Properties of Distance function:—

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

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

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



The following are the three distance measures:—

- Euclidean Distance:
  is defined as => $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

- D_4 distance (Manhattan distance or City block distance)

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

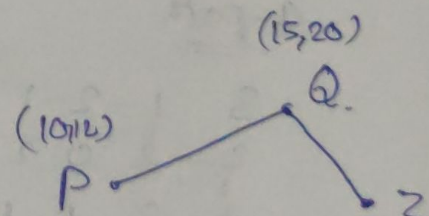
		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

- D_8 distance. (chessboard distance)

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

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

Q) Find the three distance measures of P & Q.



Sol^n. $D_e(p,q) = \sqrt{(20-12)^2 + (15-10)^2}$

$= \sqrt{64+25}$

$= \sqrt{89}$

$\simeq 9$ Ans

$D_4(p,q) = |10-15| + |12-20|$

$= 5 + 8 = 13$ Ans

$D_8(p,q) = \max(5,8)$

$= 8.$ Ans

Image enhancement in Spatial Domain (x, y Domain)

1) Point transformation
2) Histogram Equalisation
3) Mask processing

Process equation.

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

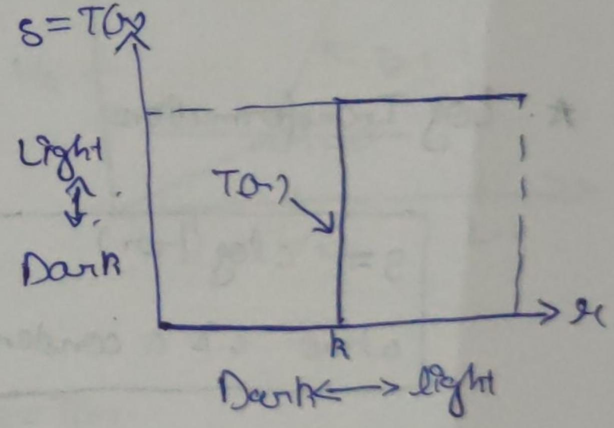
OR, $s = T(r)$

$g(x,y)$: o/p image.
$f(x,y)$: i/p image. Intensity.
T: Transformation function.

~~Soft and Hard Threshold~~

★ Thresholding function.

↳ Limiting the $T(r)$ to produce only two level (binary) image.



★ Point Processing techniques

↳ Approaches whose results ~~only~~ depend only on the intensity at a point sometimes. Ex: Contrast streching.

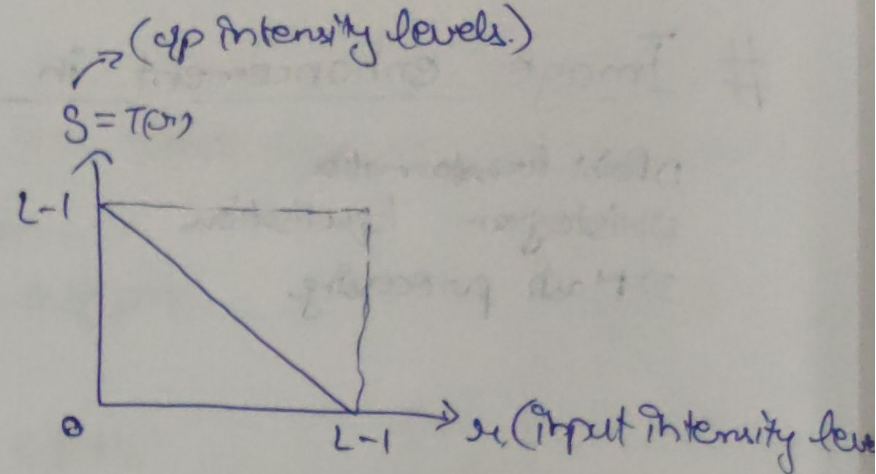
★ Neighborhood Processing techniques

↳ Approaches whose results depend on neighbour pixel as well. Ex: Averaging the Pixel values.

Converting to B&W image.

Basic Intensity Transformation Functions.

★ Image Negatives.



$$\boxed{\begin{aligned} S &= T(r) \\ &= L-1-r \end{aligned}}$$

• Reversing the intensity levels.

Ex: Apply image negative to below Image.

Soln.

$$\begin{bmatrix} 220 & 245 & 160 \\ 245 & 200 & 180 \\ 160 & 120 & 100 \\ 20 & 80 & 60 \end{bmatrix} \xrightarrow[\text{neg}]{\text{Image}} \begin{bmatrix} 35 & 10 & 95 \\ 10 & 55 & 75 \\ 95 & 135 & 155 \\ 235 & 175 & 195 \end{bmatrix}$$

r - matrix

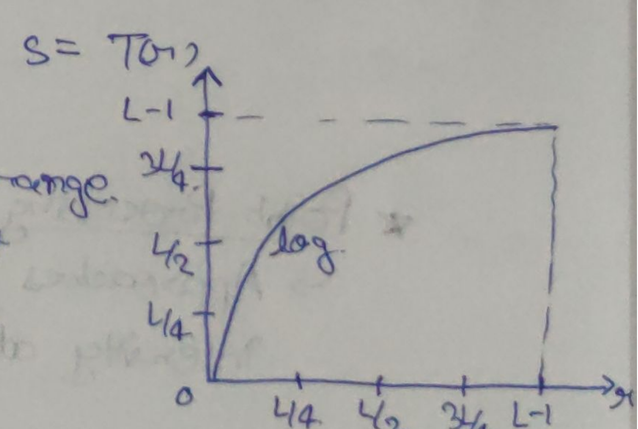
$S = 256 - 1 - r$

↳ for grayscale image.

★ Log Transformations

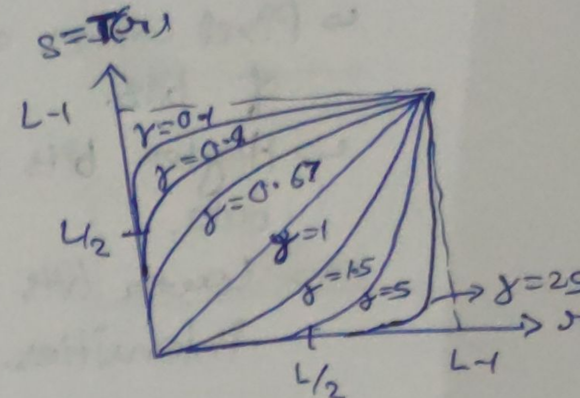
$$\boxed{\begin{aligned} &s = c\log(1+r) \\ &\text{where } c \text{ is a constant} \end{aligned}}$$

• This transformation maps the narrow range of low intensity values in the input into a wider range of o/p levels



★ Power-Law (Gamma) Transformation

$$\boxed{s = c r^{\gamma}}$$



- Power-law curves with fractional values of γ map a narrow-range of dark i/p values into a wider range of o/p values, with opposite being true for higher values.

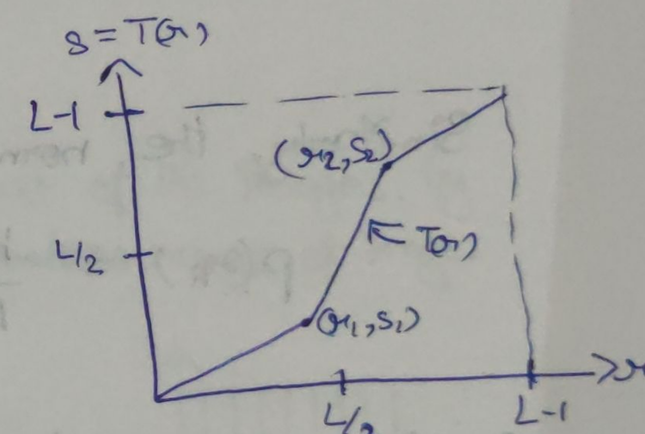
Piecewise Linear Transformation Functions

20/08/24

★ Contrast Stretching

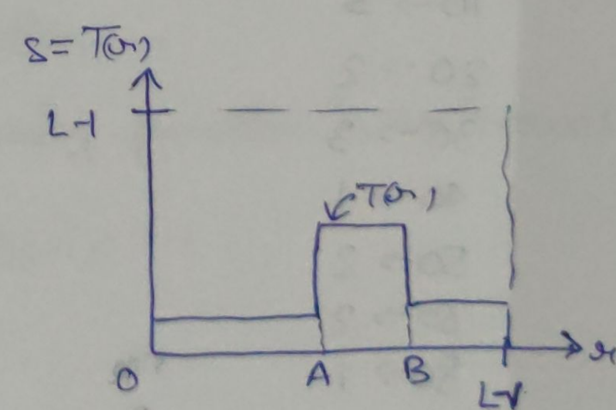
↳ Contrast stretching expands the range of intensity levels in an image. ~~so that~~

(r_1, s_1) and (r_2, s_2) control the shape of Transformation.



★ Intensity-Level Slicing

↳ To highlight a specific range of intensity in an image.



Bit-Plane Slicing

- Pixel values are integers composed of bits.
- Higher bits contains more significant data.
- Lower bits contains more subtle intensities.

Histogram Processing

$$\boxed{h(r_k) = n_k}$$

r_k is the k^{th} intensity values.
n_k is the number of pixels with intensity r_k.

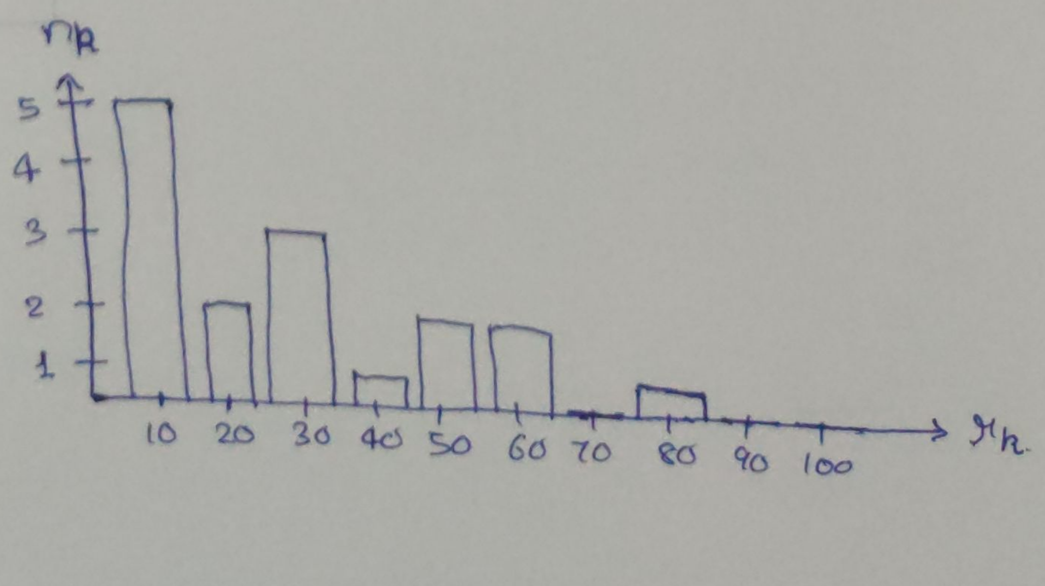
Similarly, the normalized histogram of $f(x,y)$ is,

$$p(r_k) = \frac{h(r_k)}{MN} = \frac{n_k}{MN}.$$

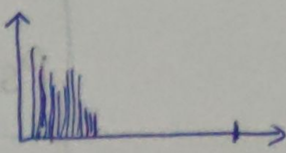
Ex: Draw histogram for $f(x,y)$ aside.

10	20	20	10
30	30	10	40
80	50	60	10
30	60	50	10

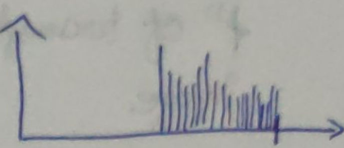
$r_k \to n_k$
$10 \to 5$
$20 \to 2$
$30 \to 3$
$40 \to 1$
$50 \to 2$
$60 \to 2$
$80 \to 1$



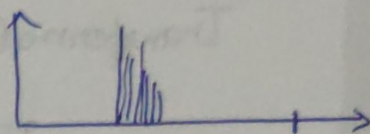
* Histogram of Dark Image.
  ↳ r_k (values) are near 0.



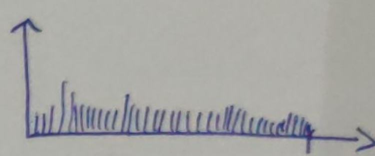
* Histogram of Bright Image.
  ↳ r_k (values) are near 255.



* Histogram of low-contrast Image.
  ↳ Centred values.

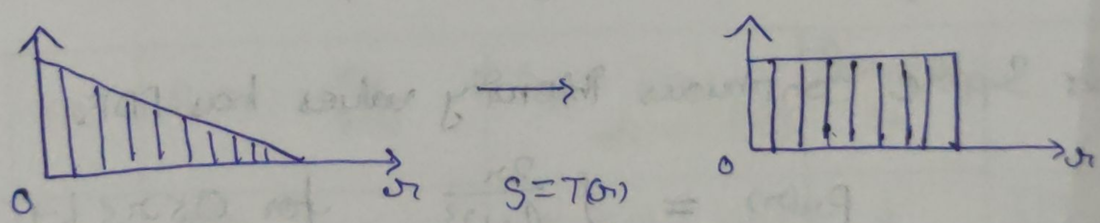


* Histogram of high-contrast Image.
  ↳ Distributed values



Histogram Equalisation

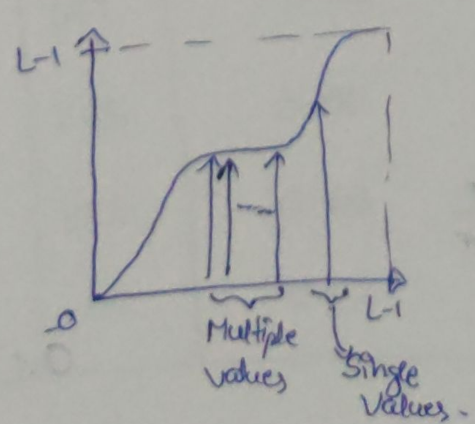
↳ Process to improve the contrast of the images. by stretching out the intensity ~~values~~ range of image.

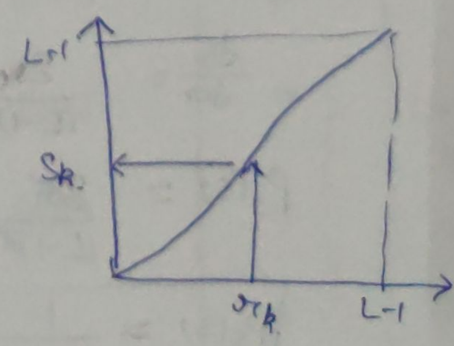


* Conditions to perform equalisation.:—

(a) $T(r)$ is a monotonic increasing function. In the interval $0 \leq r \leq L-1$.

(b) $0 \leq T(r) \leq L-1$ for $0 \leq r \leq L-1$





$$\boxed{P_S(S) = P_r(r)\left|\frac{dr}{ds}\right|}$$

Probability density f^n of transformed image.

PDF of input.

Transformation function is obtained by,

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

cumulative distribution f^n of r.

$$\frac{ds}{dr} = (L-1)\,P_r(r)$$

Now $P_S(S) = \cancel{P_r(r)} \cdot \frac{1}{(L-1)\cancel{P_r(r)}} = \frac{1}{(L-1)}$

$$P_S(S) = \frac{1}{(L-1)}$$

Hence, $P_S(S)$ is always be uniform.

Ex: Suppose continuous intensity values has PDF.

$$P_r(r) = \begin{cases} \frac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{o/w} \end{cases}$$

Sol^n.

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

$$S = \frac{1}{(L-1)}\,\frac{\cancel{2}\,r^2}{\cancel{2}} = \frac{r^2}{(L-1)}$$

$$\frac{ds}{dr} = \frac{2r}{(L-1)}$$

$$P_s(S) = \frac{\cancel{2r}}{(L-1)^{\cancel{2}}} \times \frac{\cancel{L-1}}{\cancel{2r}}$$

$$\boxed{P_S(S) = \frac{1}{(L-1)}} \qquad 0 \le S \le L-1$$

For Discrete Values

$$P_r(r_k) = \frac{n_k}{MN}$$

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

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

Q) Find Histogram equalisation.

3- bit image (L=8)

Img. size → 64x 64 pixels.

The intensity distribution of image is given below.

Grey levels (r_k)	0	1	2	3	4	5	6	7
No. of pixels (n_k)	790	1023	850	656	329	245	122	81

Solⁿ.

r_k	n_k	$P_r(r_k) = \frac{n_k}{MN}$	$S_k = T(r_k) = (L-1)\sum P_r(r_j)$	Round off S_k	$P_s(S_k)$
0	790	$\frac{790}{4096} = 0.19$	$7\times0.19 = 1.33$	1	$\frac{790}{4096} = 0.19$
1	1023	0.25	$(8-1)\times\cancel{3.08}(0.25+0.19) = 3.08$	3	$\frac{1023}{4096} = 0.25$
2	850	0.21	4.55	5	$\frac{850}{4096} = 0.21$
3	656	0.16	5.67	6	$\frac{656+329}{4096} = 0.24$
4	329	0.08	6.23	6	
5	245	0.06	6.65	7	$\frac{245+122+81}{4096} = 0.11$
6	122	0.03	6.86	7	
7	81	0.02	7	7	

Spatial Filtering

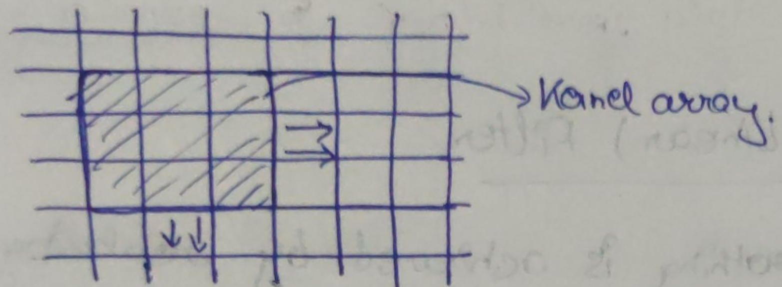
↳ Sum-of-products operation betn an image 'f' and a filter kernel 'ω'.

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

Filter value → $\omega(s,t)$; Image pixel value → $f(x+s, y+t)$

where, $x = 0, 1, \ldots, M-1$
$y = 0, 1, \ldots, N-1$

↳ Kernel is an array whose size defines the neighborhood of operation.



Zero padding is ~~the~~ to pad the neighbours of image pixels with desired no. of zero, when kernel size is more than the ~~image~~ targed image range.

Spatial Convolution:

$$(\omega \star f)(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \omega(s,t)\, f(x-s, y-t)$$

The kernel array is convolved over the image matrix. Each overlapping matrix values are multiplied with each other and the solution is stored in the central element: $f(x,y)$.

$$\frac{M-K+2P}{S}+1$$

Spatial Correlation

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

Same as spatial convolution. Just the kernel array is flipped 180° vertical and horizontal.

Smoothing Spatial Filters

- Smoothing (or averaging) spatial filters are used to reduce sharp transitions in intensity.
- It helps in noise reduction.
- It reduces the irrelevant details i.e. pixel regions that are smaller with respect to the size of the filter kernel.

Order Static (Non- Linear) Filter

- Here ~~smoothing~~ smoothing is achieved by replacing value of centre pixel with value determined by ranking pixels contained in the region encompassed by the filter.

- One such example is Median Filter.
  As name suggests, it replaces the centre pixel value with the median of encompassed pixels by filter.
- It removes the salt-and-pepper noise.
- cv2.blur() → average of encompassed pixel values.
- cv2.medianBlur() → median of encompassed pixel values.

Sharpening (Highpass) Spatial Filters

- Based on first and second order derivatives.

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

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

* Laplacian Operator

$$\boxed{\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}}$$

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

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

Laplacian kernel => $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

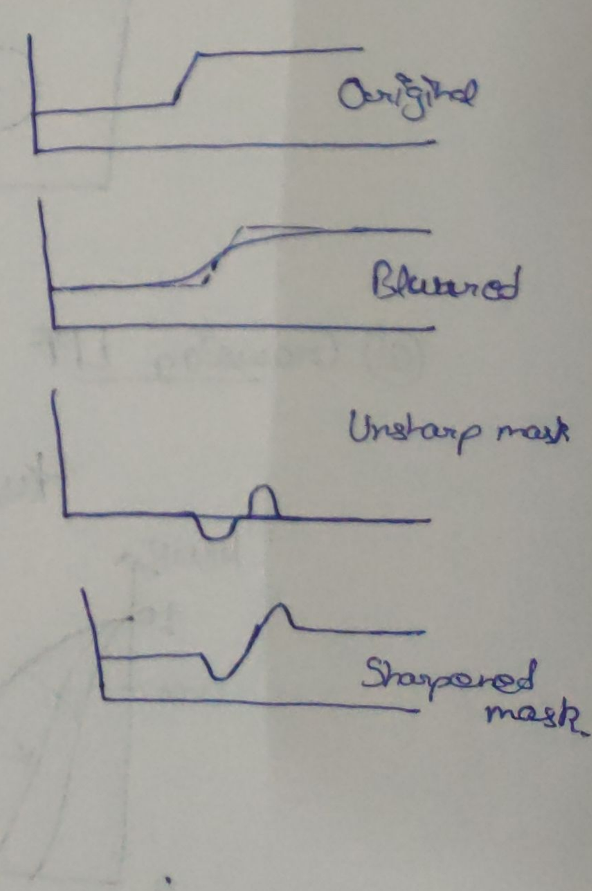
Way to use laplacian kernel :—

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

Unsharp Masking and Highboost filtering

Process of unsharp masking:—

1) Blur the original image.
2) Subtract the blurred image from the original [This results to mask].
3) Add mask to the original.



$g_{mask}(x,y) = f(x,y) - \bar{f}(x,y)$

$g(x,y) = f(x,y) + k\, g_{mask}(x,y)$

When $k=1$, we have unsharp mask.

When $k>1$, -> highboost filtering.

(a) Robert-cross gradient operator => $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

(b) Sobel operator => $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

Horz. Vert.

Image Smoothing in Frequency Domain [Once check these steps in web for more points]

① Convert spatial domain to freq. Domain by fourier transform.

② Multiply the Filter and Image in freq. Domain.

③ Retrieve back to spatial Domain using Inverse fourier transform.

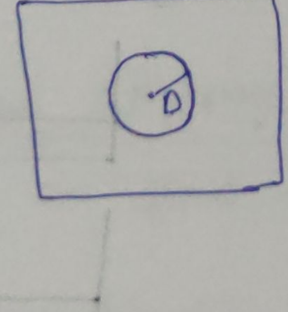
IMAGE SMOOTHING USING LOW-PASS FREQ DOMAIN FILTERS

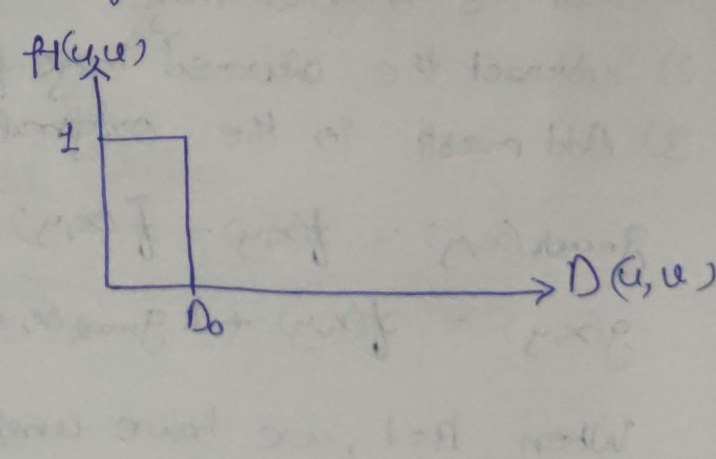
- Used to achieve smoothing.
- Three types of LPF : Ideal (sharp filter), Butterworth, Gaussian (smooth)
- The Butterworth filter provides the transition between the two extremes, just by changing it's order.

ⓐ Ideal LPF (ILPF)

- 2-D LPF that passes all freq. within a circle of radius, without attenuation.

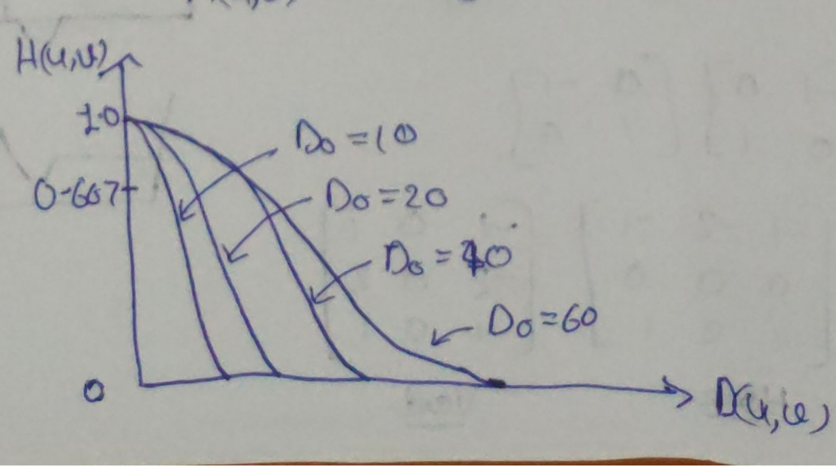
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$





ⓐ Gaussian LPF (GLPF)

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



D_0 is the cutoff frequency, ~~which~~ also the measure of spread around the centre.

ⓐ Butterworth LPF (BLPF)

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

where, n is the order of BLPF

- Higher values of n → ILPF
- Lower values of n → GLPF