NOVEMBER

22 DIP 23 24

326-39 THURSDAY 327-38 FRIDAY 328-37 SATURDAY

7

Pubs in England open 24 hours as new licensing law takes effect 2005

8

Chapter:

9

(Histogram Specification/matching topic টা বাদ)

- 3 → Lecture Slide: 4, 5, 6, 7

10

- 6 → Lecture: 15 (HSI model টা বাদ full. বাকিটা আছে।)

11

- 8 → Lecture: 13, 14

12

- 9 → Lecture: 11, 12 (thinning & thickening এর আগে পর্যন্ত)

1

- 10 → Lecture: 8

25

2

329-36 SUNDAY

3

Lionel Messi becomes the UEFA Champions League all-time top scorer 2014

4

4'2

Chapter: 3 (Intensity Transformations)

Lecture : 4

- Enhancement
- Intensity transformation functions
  - → Linear
  - → Logarithmic
  - → Power law
- Piecewise-Linear Transformation function
  - → Contrast stretching
  - → Intensity-level slicing
  - → Bit-plane slicing

Image Enhancement:

* Process/manipulate an image so that the result is more suitable than the original image for a specific application.

Methods:

⇒ Spatial Domain ← We will only learn about this

⇒ Frequency Domain

⇒ Combination Methods

Spatial Domain Process:

Spatial Domain:
Direct manipulation of pixels in an image

Types of img enhancement operations:

① Point/Pixel operation

A এর (x, y) এর উপর Perform করে B এর (x, y) পেলাম।

② Local operation

A এর (x. y) এর neighbours এর উপর perform করে B এর (x, y) পেলাম

Spatial Filters: by mask, kernel, template or window

Intensity Transformation: Point operation

③ Global operation (Frequency Domain)

A এর full ~~img~~ image এর উপর কাজ করে B এর (x, y) পেলাম

* input, output image এর size same হবে।

Intensity transformations functions:

- ① Basic:
  - → Linear
  - → Logarithmic
  - → Power-law
- ② Piecewise Linear
  - → Contrast stretching, thresholding
  - → Gray-level slicing
  - → Bit-plane slicing

$$\overset{s}{g(x,y)} = T\,[\overset{r}{f(x,y)}]$$

T → Gray-level/ intensity transformation/ mapping function

$\therefore s = T(r)$

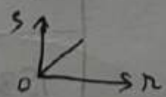
r = Grey level of f at (x,y)

S = Grey level of g at (x,y)

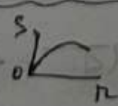
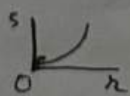
T = A function that maps r to s.

* 8 bit img → $k=8$ So, $L-1 = 2^8 - 1$
$= 256 - 1$
$= 255$

Linear Functions:

- Negative  inverse করে দেয় $\therefore S = (L-1) - r$
- Identity  $s = r$ (input, output same)

Logarithmic Functions:

- Log 
- Inverse-log 

Power-Law functions:

- nth power 
- nth root 

(Linear)

Negative Transformation:

Pixel by Pixel কাজ

8-bit image

$L = 2^8$

$= 256$

$L-1 = 256-1$

$= 255$

Input

100	110	90	95
98	140	195	135
89	90	88	85
102	105	99	115

$f(x,y)$

We know,

$$\boxed{S = (L-1) - r}$$

so, For first pixel $r = 100$

$\therefore S = 255 - 100$

$= 155$

For second pixel $r = 110$

$\therefore S = 255 - 110$

$= 145$

For third pixel $r = 90$

$\therefore S = 255 - 90$

$= 165$

Output

155	145	165	160
157	115	110	120
166	165	167	170
153	150	156	140

$g(x,y)$

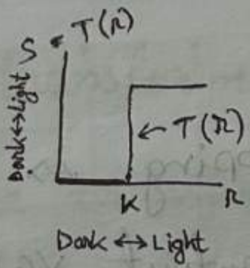
এভাবে each pixel count হবে then output image আসবে।

Thresholding:

* যেকোনো gray level image কে binary image-এ নিতে thresholding use করা হয়।
  Threshold value থেকে বড় value গুলোকে 1 এবং ছোট value গুলোকে 0 করে ফেলবো।

* contrast enhancement এর জন্য thresholding use করা হয়।

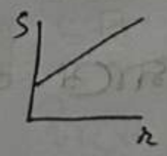
* Function:



$S = T(r)$

(Linear) Intensity Scaling:

$$S = T(r) = a \cdot r$$



Log Transformation:

$$S = c * \log(1+r)$$

* input image এর lower amplitudes গুলোর grey level কে expand করে।

* input image এর higher amplitudes গুলো grey level কে compressed করে।

* Also called "dynamic-range compression/expansion".

Power-Law transformation :

Gamma Transformation

$S = C * r^{\gamma}$ ⇐ c & γ constant

c = 1 ধরে নেয় usually by default

$\gamma < 1$ ⇒ the mapping is weighted toward brighter output values.

$\gamma = 1$ ⇒ the mapping is linear.
(default)

$\gamma > 1$ ⇒ the mapping is weighted toward darker output values.

* MRI তে Power-Law use করা হয়, γ এর value বাড়িয়ে কমিয়ে detailing বুঝা হয় MRI তে।

Piecewise Linear Transformation Functions:

→ Advantage: Some important transformations can be formulated only as a piecewise function.

→ Disadvantage: Their specification requires more user input than previous transformations.

* 3 types Piecewise transformation:

1. Contrast Stretching:

→ এই contrast stretching এর main কাজ হচ্ছে dynamic range কে increase করা।

$$* \; I_{new} = (I - Min) \frac{NewMax - NewMin}{Max - Min} + NewMin$$

↳ I = Input Intensity

NewMax = 255, NewMin = 0

max = 150, min = 80

I_{New} = Output Intensity after contrast streching

If. input image: 8bit have range [80-150]

After contrast streching the new range becomes: [0 - 255]

then

Exercise:

110	120	90	130
91	94	98	200
90	91	99	100
82	96	85	90

8-bit image.

So the range is (0-255)

Given, $m = 95$ [threshold]

Output image:

255	255	0	255
0	0	255	255
0	0	255	255
0	255	0	0

95 থেকে ছোট value গুলো 0 হয়ে যাবে। এবং 95 থেকে বড় value গুলো ২৫৫ হয়ে যাবে। যদি 95 থাকতো সেই ক্ষেত্রে 0 হয়ে যেতো।

2. Gray / Intensity Level Slicing:

↳ Highlights a specific range of Intensities levels in an image.

Example: যদি, (100 → 200) এমন একটা range দেয়া আছে। এখানে, (100 → 200) এর মধ্যের সব pixel গুলোর 255 (white) করে দিবো। বাকি যা pixel আছে সেগুলোর 0 (black) করে দিবো।

3. Bit - Plane Slicing:

MSB

100110010

LSB

* Far Left side is MSB.
* Far Right side is LSB.

↳ Three main goals are:

① Converting grey level image to a binary image.

② compression

③ Enhancing the image by focusing.

* The n^{th} plane in the pixels are multiplied by the constant 2^{n-1}

Exercise:

value গুলো decimal -এ দেয়া থাকলে binary তে convert করে নিতে হবে

8-bit image:

10100111	10000101	01101111
10010000	10001100	10000111
10011111	10011010	10010100

* bit-plane slicing করলে 8 টি matrix পাওয়া যাবে। LSB এর position থেকে matrix create শুরু করতে হবে।

bit-plane Slicing:

matrix-1

1	1	1
0	0	1
1	0	0

matrix-2

1	0	1
0	0	1
1	1	0

matrix-3

1	1	1
0	1	1
1	0	1

matrix-4

0	0	1
0	1	0
1	1	0

matrix-5

0	0	0
1	0	0
1	1	1

matrix-6

1	0	1
0	0	0
0	0	0

matrix-7

0	0	1
0	1	0
0	0	0

matrix-8

1	1	0
1	1	1
1	1	1

Reconstruct the image: (using 5th and 4th bit planes)

(2^{n-1}) ← used here

$0\times2^{5-1}+0\times2^{4-1}$ $=0\times2^4+0\times2^3$ $=0$	$0\times2^{5-1}+0\times2^{4-1}$ $=0$	$0\times2^{5-1}+1\times2^{4-1}$ $=0+1\times2^3$ $=8$
$1\times2^{5-1}+0\times2^{4-1}$ $=1\times2^4+0$ $=16$	$0\times2^{5-1}+1\times2^{4-1}$ $=0+1\times2^3$ $=8$	$0\times2^{5-1}+0\times2^{4-1}$ $=0$
$1\times2^{5-1}+1\times2^{4-1}$ $=1\times2^4+1\times2^3$ $=16+8$ $=24$	$1\times2^{5-1}+1\times2^{4-1}$ $=1\times2^4+1\times2^3$ $=24$	$1\times2^{5-1}+0\times2^{4-1}$ $=1\times2^4+0$ $=16$

Final Result: (Enheched image) Also, it is a compressed image.

0	0	8
16	8	0
24	24	16

Slide exercise:

2	1	2	1	0
7	1	4	3	2
2	4	1	3	7
1	3	4	6	3
1	4	1	3	4

→ 3-bit image

→ 3-bit image এর highest value হয় 7. এবং lowest 0.

Digital Negative:

$7-2 = 5$	$7-1 = 6$	5	6	7
$7-7 = 0$	6	3	4	5
5	3	6	4	0
6	4	3	1	4
6	3	6	4	3

Digital negative বললে max value যা হয় তা থেকে ঐ certain pixel এর value টা বিয়োগ করে then ঐ position-এ বসায়ে দিবে।

Bit-plane:

* Firstly decimal value গুলোকে binary তে নিয়ে একটা matrix বানাবো। যেহেতু 3-bit image so binary value গুলা 3 bit এর হবে।

* Then ঐ matrix থেকে bit-plane করলে new 3 টা matrix পাবো। এগুলোই হয়ে গেলো bit-plane. Reconstruction করার দরকার নেই এখানে কারণ বলে নাই করতে।

Thresholding with T=3:

* Pixel এর value 3 বা 3 থেকে ছোট হলে 0 (black) বসাবো।
* Pixel এর value 3 থেকে বড় হলে 7 (white) বসাবো।

0	0	0	0	0
7	0	7	0	0
0	7	0	0	7
0	0	7	7	0
0	7	0	0	7

* 3-bit image তাই max value টা 7 হলো।
* যদি 1-bit / binarized image বলতো তখন 7 এর জায়গায় 1 হতো।

Intensity slicing with both approaches.

when $a = 3$, $b = 5$; up the level to 5 :

First approach :

* range এর মধ্যে না থাকলে 0 করে দিব।
* range এর মধ্যে থাকলে 5 করে দিব।

0	0	0	0	0
0	0	5	5	0
0	5	0	5	0
0	5	5	0	5
0	5	0	5	5

Second approach:

* range এর মধ্যে না থাকলে যা আছে তাই বসাবে।
* range এর মধ্যে থাকলে 5 করে দিব।

2	1	2	1	0
7	1	5	5	2
2	5	1	5	7
1	5	5	6	5
1	5	1	5	5

Histogram

Lecture : 5

Histogram: Histogram is a graphical representation showing a visual impression of the distribution of data.

Image Histogram: Image histogram is a type of histogram that acts as a graphical representation of the lightness/color distribution in a digital image. It plots the number of pixels for each value.

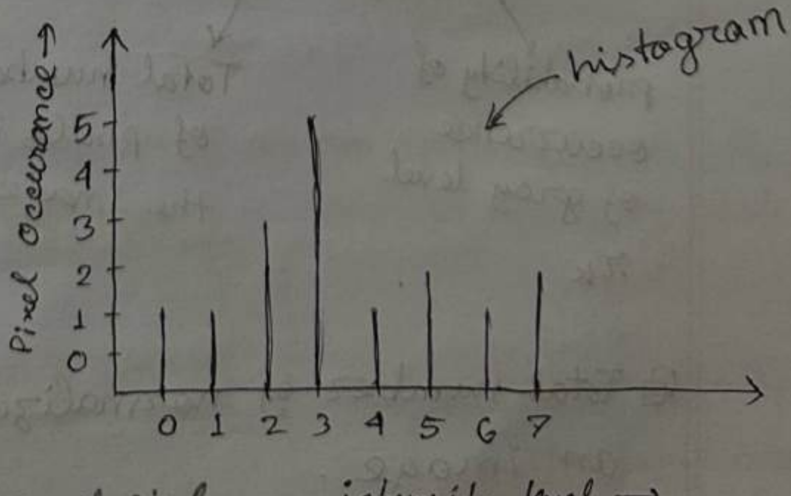
Example:

a 3-bit 4×4 image. So, $L = 2^n = 2^3 = 8 \rightarrow (0-7)$ ← Intensity Level

An image

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

Position	0	1	2	3	4	5	6	7
h	1	1	3	5	1	2	1	2



$$\sum_{x=0}^{2^b-1} h(x) = \underbrace{\text{area of the image}}$$

↳ এটার মানে total no. of pixels

- Gray level যদি $[0, L-1]$ এই range এর হয় then,

$$h(r_k) = n_k$$

Here,

r_k = k^{th} gray level

n_k = no. of pixels with having gray level r_k

- different image but no of pixel same also they are gray level images then তাদের histogram কখনো same হবে।

- **Histogram Normalization:**

$$p(r_k) = \frac{n_k}{n} \quad \text{where } n = \sum_k n_k$$

n_k → certain 1টা value এর Total no. of pixel

$p(r_k)$ → probability of occurrence of gray level r_k

n → Total number of pixels in the image

- Total number of normalized histogram is 1 for an image.

☐ No ~~idea~~ indication of location of pixels.

☐ Can we reconstruct image from histogram?

Ans: No.

Color img Histogram:

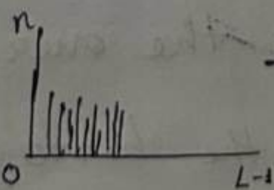
2 ways:

① convert color img to gray scale. Then display histogram of gray scale.

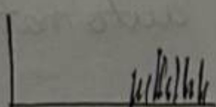
② 3 histograms (R, G, B).

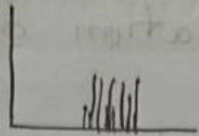
1 for Red, 1 for Green, 1 for Blue.

Four basic type of imgs and their corresponding histograms:

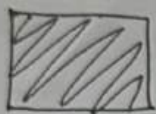
①  ← Dark img 

→ concentrated on the low side of the gray scale.

②  ← Bright img  → concentrated on the high side of the gray scale.

③  ← Low contrast img  → narrow and centered toward the middle of the gray scale.

Pic টা ↙ ভুল হয়ে গেছে

④  ← high contrast img  → covers broad range of the gray scale & the distribution of pixels is not too far from uniform with very few vertical lines being much higher than the others.

Q. Why is the img histogram important?

Ans: 1) Histogram এর shape টা কিছু key info provide করে।

- The overall quality of the img.
- The possibility for enhancement.

2) Visual inspection of the ~~hig~~ histogram is a fast way to analyze the overall img quality.

3) By processing the histogram an img can be quickly and automatically improved.

Scanned with CamScanner

Uses of Histogram:

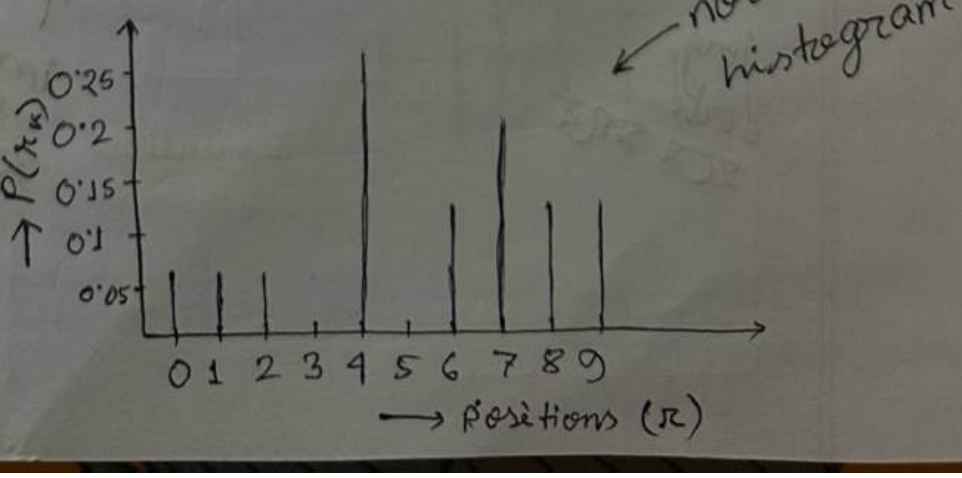
1) whether img was scanned properly or not.
2) idea about tonal distribution
3) to improve appearance, Histogram equalization can be applied.
4) tells us about the objects of the img.
5) to select threshold value for obj detection.
6) can be used for managed segmentation.

Normalized img Histogram:

position	0	1	2	3	4	5	6	7	8	9
$h(r_k)$	1	1	1	0	4	0	2	3	2	2
$p(r_k)$	1/16	1/16	1/16	0	4/16	0	2/16	3/16	2/16	2/16

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

An img



Histogram processing techniques:

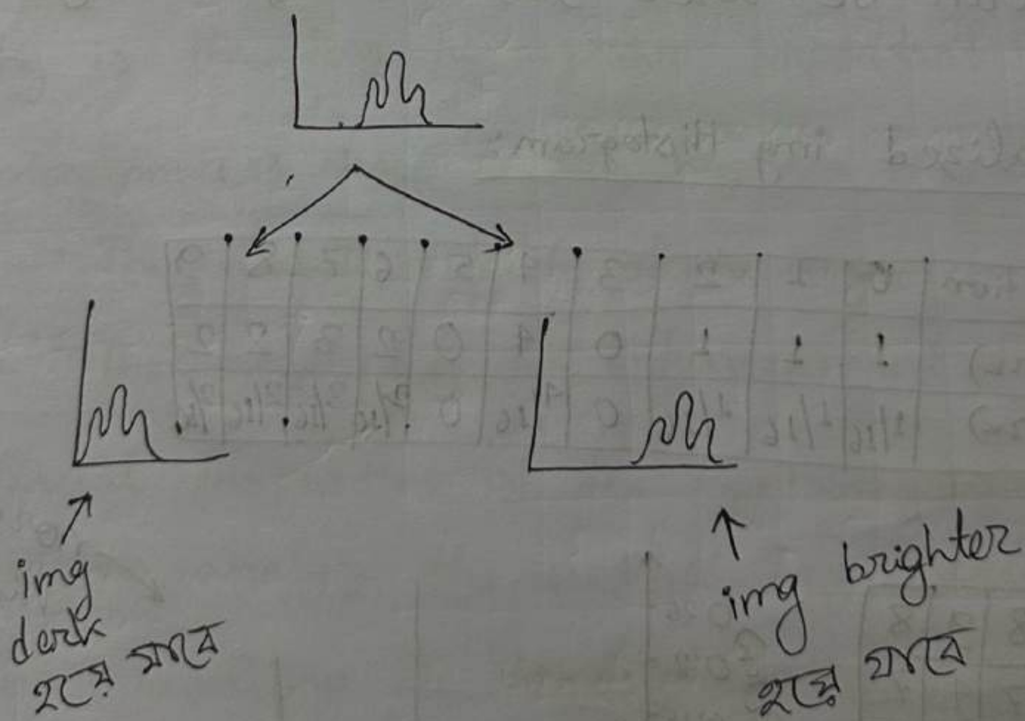
① Histogram Sliding

② Histogram Stretching

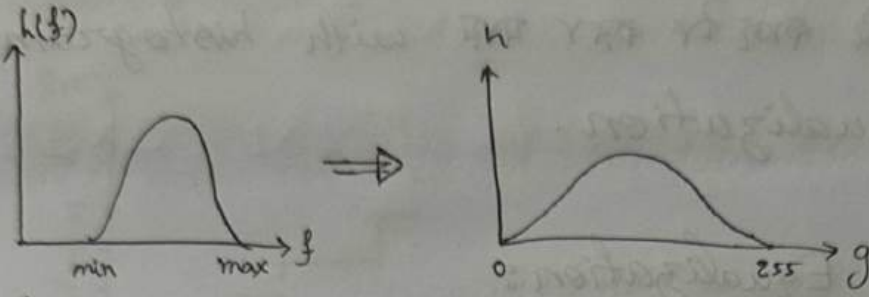
③ Histogram Equalization

Histogram Sliding:

→ simply shifted towards rightwards or leftwards.



Histogram Stretching:



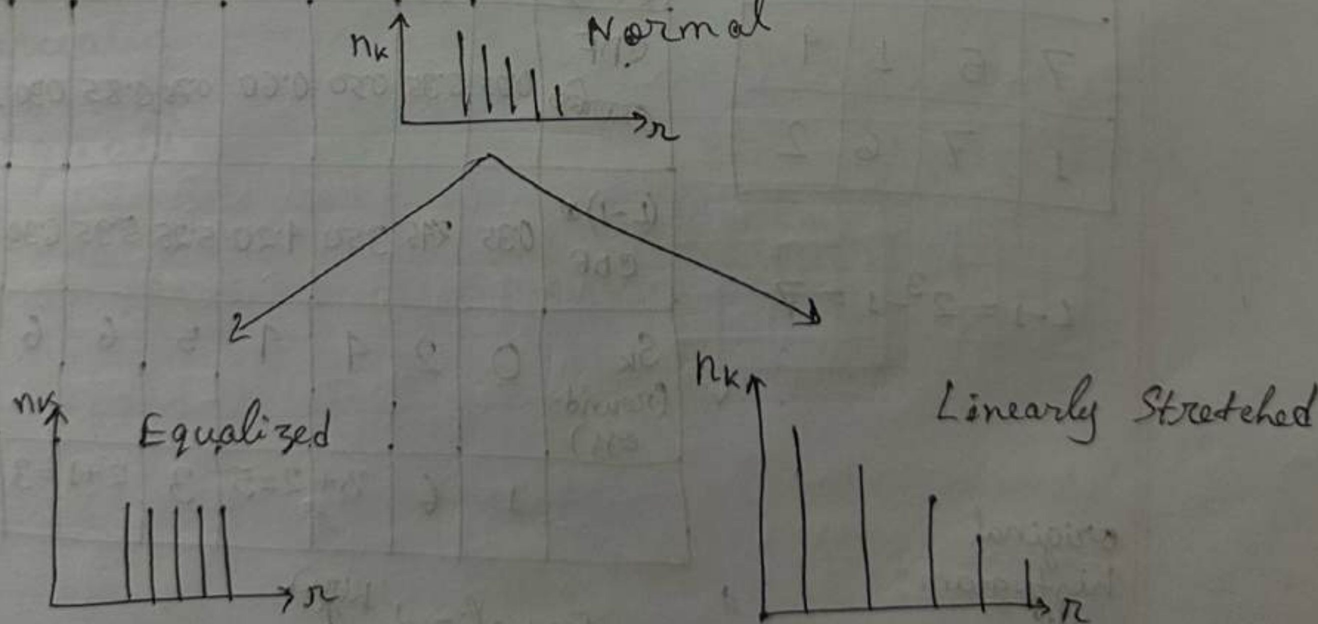
Rescale to [0, 255] range

Slide histogram down to 0

$$g = \frac{255\,(f - \min)}{\max - \min}$$

Normalized histogram to [0, 1] range.

Stretch করলে enhance হয়ে যায়।



এটা best

☐ most of the time we need a flat histogram.
↳ এই কাজটি করা যায় with histogram equalization.

Histogram Equalization:

Target → contrast enhance করা

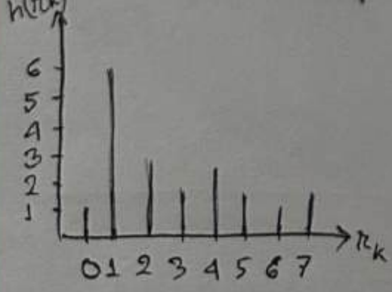
0	1	2	1
3	1	4	5
1	4	2	3
7	5	1	4
1	7	6	2

$L-1 = 2^3 - 1 = 7$

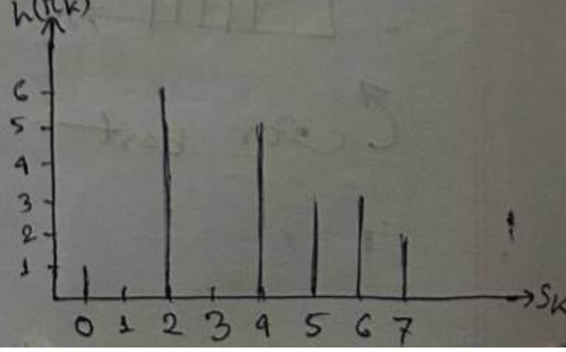
r_k	0	1	2	3	4	5	6	7
$h(r_k)$	1	6	3	2	3	2	1	2
PDF ($p(r_k)$)	1/20 =0.05	6/20 =0.30	3/20 =0.15	2/20 =0.10	3/20 =0.15	2/20 =0.10	1/20 =0.05	2/20 =0.10
CDF ক্রমযোজিত	0.05	0.35	0.50	0.60	0.75	0.85	0.90	1.00
$(L-1)*$ CDF	0.35	2.45	3.50	4.20	5.25	5.95	6.30	7.00
S_k (round-off)	0	2	4	4	5	6	6	7
	1	6	3+2=5		3	2+1=3		2

Total = 20

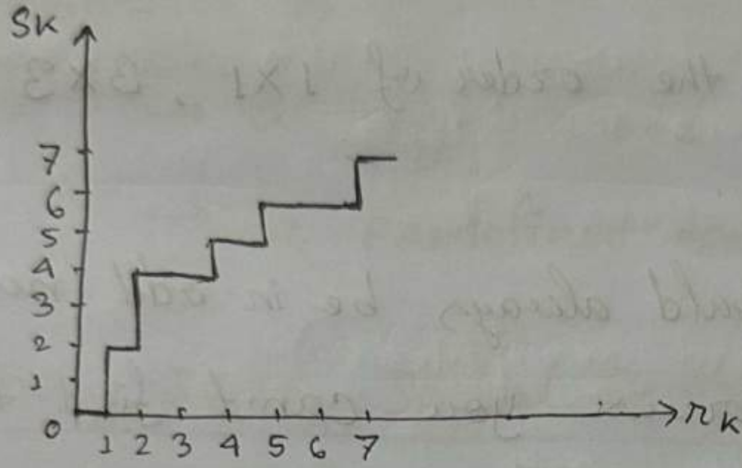
original histogram:



Equalized histogram:

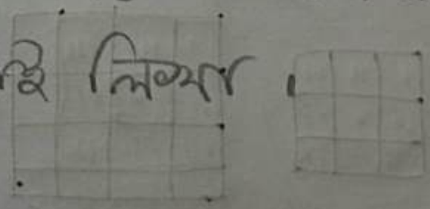


Transformation function curve :



Histogram Equalization Algorithm:

usually আসে না। যদি আসেও just equalization এর process টা লিখলেই enough. Slide-এও তাই লিখা।



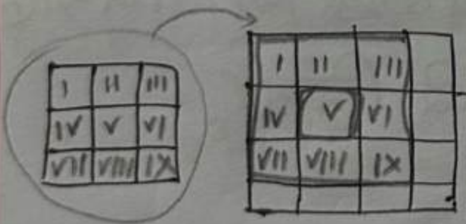
What is kernel / mask? **Lecture: 6**

Ans: 2-dimensional matrix. The mask is usually of the order of 1×1, 3×3, 5×5, 7×7.

A mask should always be in odd number becuse otherwise you can't find the mid of the mask.

Also known as filter / mask / window / box / kernel.

Convulotion Process:



element wise গুণ হয়ে হয়ে যাচ্ছে।

প্রতিটির pixel এর উপর input img এর project করে করে value গুলোর বের হবে। And each pixel এর জন্য করাটার করার পর finally একটার output img পাওয়া যাবে।

Spatial Filtering:

consists of :

→ a neighborhood

→ a predefined operation

Filtering in the Spatial Domain:

2 types : ① Smoothing Spatial filter [Low Pass].

② Sharpening Spatial filter [High Pass]

Q. Why filters are used ?

Ans → For blurring and noise reduction

→ For edge detection and sharpness.

→ For enhance img

→ For extract info from img

→ For detecting patterns.

Smoothing Spatial Filters:

Use: → for blurring and → noise reduction.
→ remove fine details.

Types of smoothing filter:

1. Standard avg / Mean
2. Weighted avg

$\Big\}$ linear

3. Median filter $\}$ Order Statistic (Non-linear)

① Standard Avg / Mean Filter: (Linear)

→ kernel এর সবগুলোর element এর value same হতে হবে এবং value গুলোর যোগ করে Total 1 আসতে হবে।

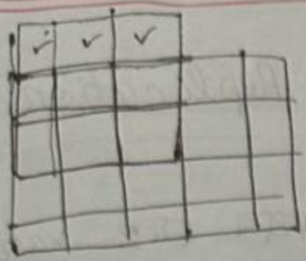
basic mean filter →

1	1	1
1	1	1
1	1	1

যেহেতু 3×3 img তাই এখানে $\frac{1}{9}$ গুন করে দিলে kernel এর each pixel value-ও same থাকে plus যোগফলও 1 হয়।

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

এখানথেকে 1/9 common নিয়ে সাজানোও লেখা যাবে

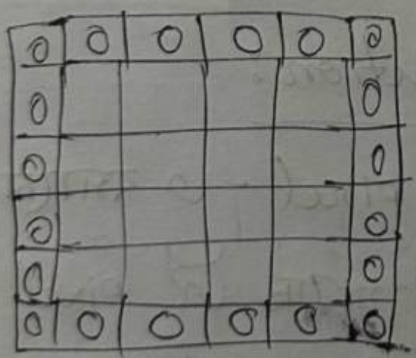


✓ – এই জায়গা গুলোয় img এর value নাই। সেক্ষেত্রে padding করে নিতে হবে।

Padding 3 types :

① <u>Zero padding</u> : (mostly used)

img এর চারপাশে দিয়ে 0 বসাবো। এতে করে img এর চারপাশে একটা black border আসবে।



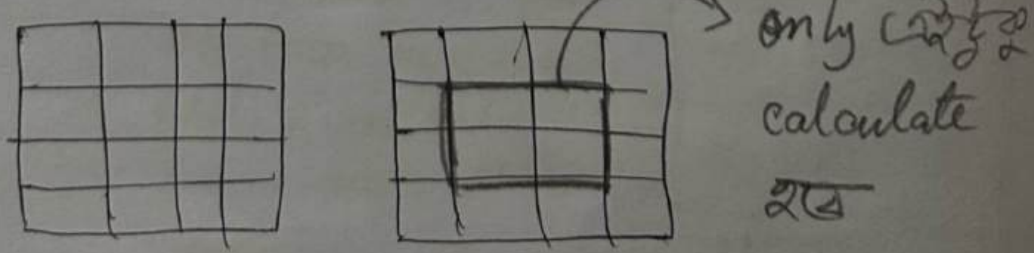
② Border padding / Pixel Replication :

img এর border বরাবর যেই value গুলা আছে সেগুলা বসায় দিব।

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

③ Discard solution :

- মানে যেই pixel-এ বসালে kernel বাইরে চলে যাচ্ছে ঐ pixel গুলার calculate-ই করবো না। Discard করে দিবো।



☐ What does Mean or Box filter do?

Ans: ① Replaces each pixel with an avg of its neighborhood

② Achieve smoothing effect. (remove sharp features).

③ Adds a 'softer' look to an image.

② Weighted Avg Smoothing Filter: (Linear)

kernel এর সবগুলোর pixel এর value same হবে না, But যোগ করলে 1 আসতে হবে।

1	2	1
2	4	2
1	2	1

যেহেতু 3x3 kernel so $\frac{1}{16}$ দিয়ে গুণ করলে element wise then সবগুলোর যোগফল 1 হয়।

$1/16$	$2/16$	$1/16$
$2/16$	$4/16$	$2/16$
$1/16$	$2/16$	$1/16$

এখান থেকে $1/16$ common নিয়েও লিখা যায়।

Weighted filter হচ্ছে mainly Gaussian Filter.

③ Median Filter: (Non-linear)

→ কোনো আলাদা kernel use করা হয় না।

→ image এর ভিতরেই kernel size এর position ধরে ধরে কাজ করা লাগে।

Example:

110	120	90	130
91	94	98	200
90	95	99	100
82	96	85	90

→

110	120	90	130
91	95	98	200
90	95	99	100
82	96	85	90

Sort করবো ছোট থেকে বড়:

90, 90, 91, 94, 95, 98, 99, 110, 120

Same way তে, যদি min filter বলে then sort করে minimum value টা বসাবো।

যদি max filter বলে then sort করে maximum value টা বসাবো।

⊞ সাদা-কালো dots থাকলে image-এ then ওগুলোকে বলে 'Salt and Pepper' noise.

এই 'Salt and Pepper' noise দূর করার জন্য median filter use করা best.

Cross co-relation and templete matching:

Slide এর মধ্যে সুন্দর মতো দেয়া আছে।

Sharpening Spatial Filter:

Lecture: 7

→ seek to highlight fine details

→ Remove blurring

→ Highlight edges

→ useful for emphasizing transitions in image intensity.

Spatial Derivation:

1st Derivation
2nd Derivation
] both has forward and backward

details in slide

Types of Sharpening Spatial Filters:

① Laplacian filter

↳ uses 2nd derivative of Image Enhancement

② Sobel (Gradient Operators)

↳ uses 1st derivative of Image Enhancement.

① Laplacian Filter:

4 types:

①
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$
Forward 4-neighbor

②
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$
Backward 4-neighbor

③
$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
Forward 8-neighbor

④
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$
Backward 8-neighbor

original img +	Laplacian filter applied img with Laplace filter	=	Enhanced Image

↳ Sharpening filter এর time-এই এমন ভাবে enhanced img পাচ্ছি।
But Smoothing এর time-এ just filter apply করলেই enhanced image পেয়ে যাচ্ছিলাম।

② Sobel operator:

−1	0	+1
−2	0	+2
−1	0	+1

G_x

+1	+2	+1
0	0	0
−1	−2	−1

G_y

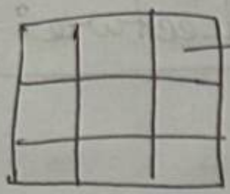
Prewitt operator: (Sobel এর মতোই অনেকটা)

−1	0	+1
−1	0	+1
−1	0	+1

G_x

+1	+1	+1
0	0	0
−1	−1	−1

G_y

 → এই একটার pixel এর জন্য sobel এর ক্ষেত্রে, একবার G_x বের করবো, একবার G_y বের করবো। Then,

$$M = |G_x| + |G_y|$$

$= \boxed{}$ → এই value টা আমরা সেটার যেয়ে ঐ certain pixel-এ বসাবো।

Chapter: 6 (Color Image Processing)

Lecture: 15

Human eye range:

$\boxed{400 \text{ nm to } 700 \text{ nm}}$

Color Model:

→ human eye combines 3 primary colors by using the 3 different types of cones to discern all possible colors.

→ colors are just different light frequencies:

red → 700 nm wavelength	
green → 546.1 nm wavelength	
blue → 435.8 nm wavelength	

→ Higher frequencies are cooler colors.

↳ 700 nm হলে red.

↳ 400 nm হলে violet.

Human Color Vision:

- Retina contain 2 types of Photoreceptors
  - ↳ Cones (color sensitive)
  - ↳ Rods (more numerous, insensitive to color)

- 6 to 7 million cones in each eye are responsible for colour vision.
  - ↳ 65% of these are sensitive to Red light.
  - ↳ 33% to Green light
  - ↳ 2% to Blue light. (Blue cones are the most sensitive.)

Primary Colors:

⇒ Primary colors of light are additive

↳ Primary Colors: Red, Green, Blue

↳ combining red + green + blue yields white.

⇒ Primary colors of pigment are subtractive

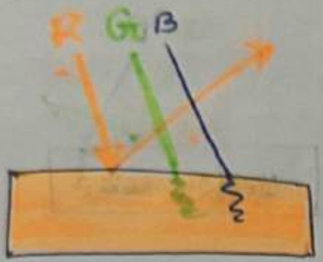
↳ Primary Colors: Cyan, Magenta, Yellow

↳ Cobining cyan + magenta + yellow yields black.

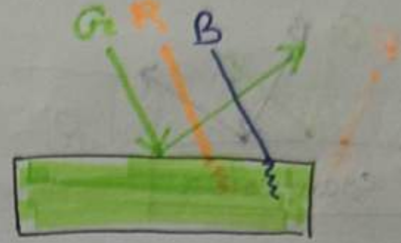
Mixture of light (Additive)	Mixture of pigments (subtrative)
↳ Active display (TV, PC)	↳ Printer

Red color বাসায় খুঁজে পাইনি ^^

Red pigment:



Green pigment:



Blue Pigment:

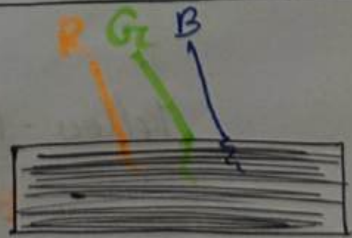
Red-Green pigment:

Black
(no light reflected)

Red-Blue pigment:

Black
(no light reflected)

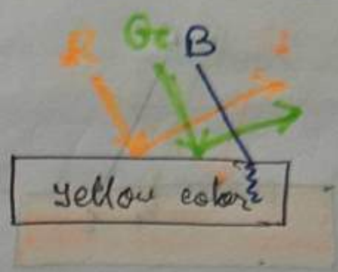
Green - Blue pigment:



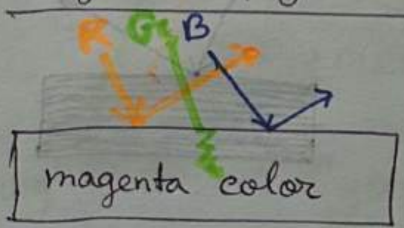
Black
(no light reflected)

Cyan pigment:

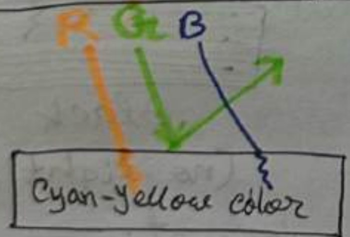
Yellow pigment:



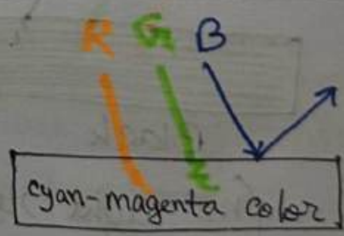
Magenta pigment:



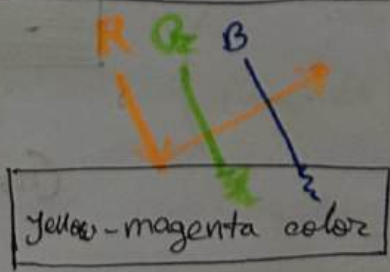
Cyan-Yellow pigment:



Cyan-Magenta pigment:



Yellow-Magenta pigment:



Primary and Secondary colors:

Additive Primaries of Light

Red + Green + Blue = White

Red + Green = Yellow

Red + Blue = Magenta

Green + Blue = Cyan

Subtractive Primaries of Pigments

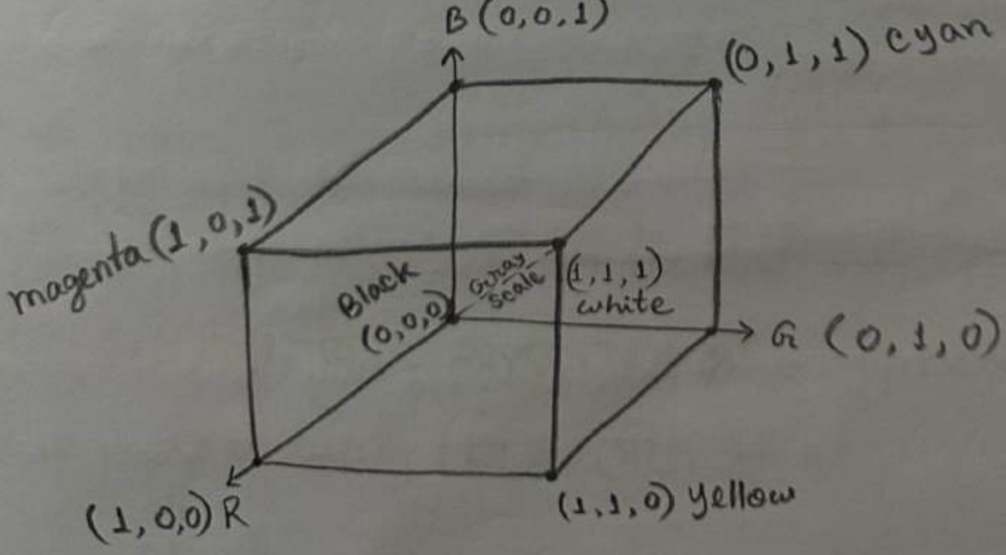
Cyan + Magenta + Yellow = Black

Cyan = White − Red

Magenta = White − Green

Yellow = White − Blue

RGB color model:



RGB 24-bit color cube

- Pixel depth = 24 bits (8 bit per plane)
- Total no. of colors = $(2^8)^3$ = 16,777,216

Light Intensity:

$$\boxed{\text{intensity} = 0.299 * \text{Red} + 0.587 * \text{Green} + 0.144 * \text{Blue}}$$

Green portion বেশি দেখে human eye এই ক্ষেত্রে

Chapter : 10 (Image Segmentation)

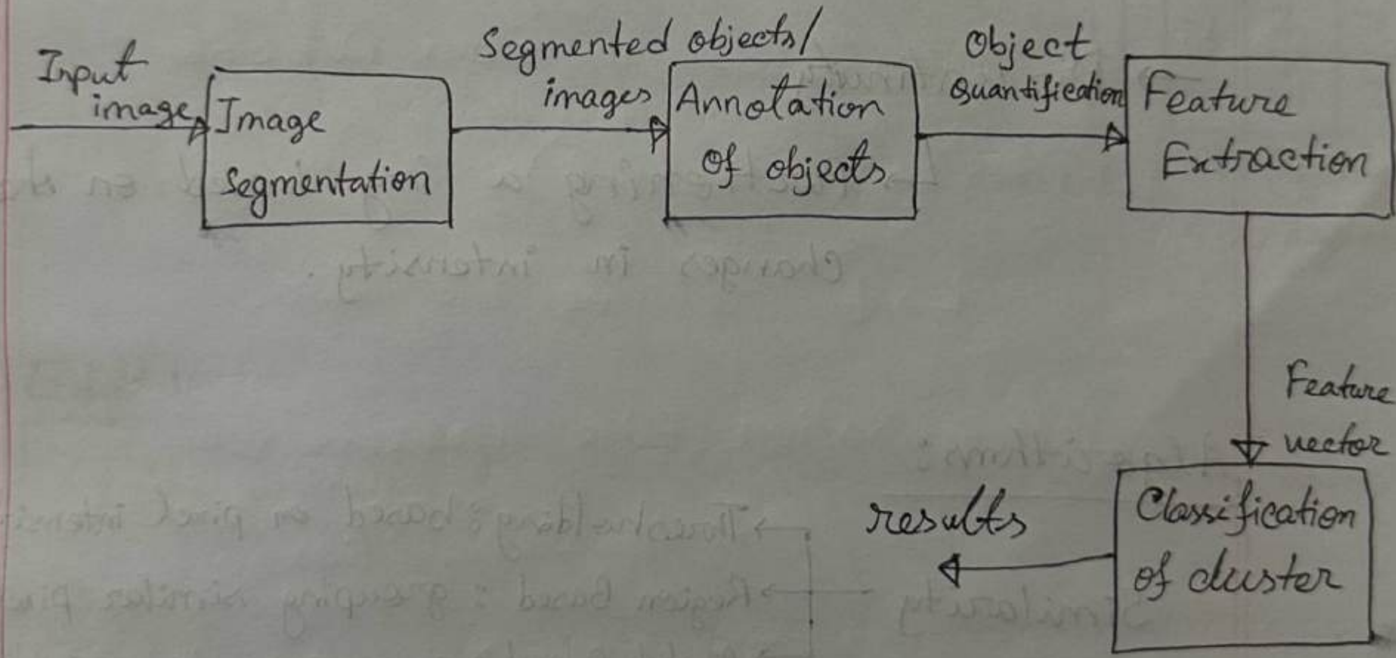
Lecture : 8

◫ What is image segmentation?

Ans: → an aspect of img processing

→ a computer vision process

→ the first step in image analysis



◫ What does img segmentation do?

⇒ একটা img কে different different way তে analysis করতে চাওয়ার জন্য আমাদেরকে segmentation process এর মধ্যে দিয়ে যেতে হবে।

Principal approaches of segmentation:

→ Similarity

↳ Partioning an img into regions that are similar according to a set of predefines criteria.

→ Discontinuity

↳ Partioning a img based on sharp changes in intensity.

Algorithms:

Similarity
- → Thresholding : based on pixel intensities
- → Region Based : grouping similar pixels
- → Match Based : comparison to a given template

Discontinuity
- → Edge based : detection of edges that separate regions from each other.
- → Watershed : find regions corresponding to local minima in intensity

Point detection:

$$R = \sum \underbrace{W_K}_{\text{img}} \overbrace{Z_K}^{\text{kernel/filter}}$$

Line detection:

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

Horizontal

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

+45° (Positive diagonal)

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

Vertical

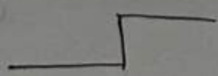
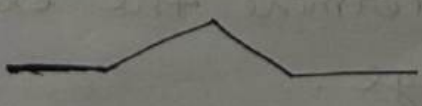
2	-1	-1
-1	2	-1
-1	-1	2

-45° (negative diagonal)

Edge:

↳ the boundary between two homogeneous regions

types:

- Step edge : 
- Ramp edge : 
- Ridge : 
- Roof : 

Edge detection:

- Identifying sudden change in image intensity

usage:

- extact imp features of an img (corners, lines, curves).

Three main steps in Edge Detection:

1. Filtering (Smoothing).

   ↳ to remove the noise.

2. Differentiation (Edge sharpening using derivatives)

   ↳ detects discontinuities

3. Detection (Thresholding)

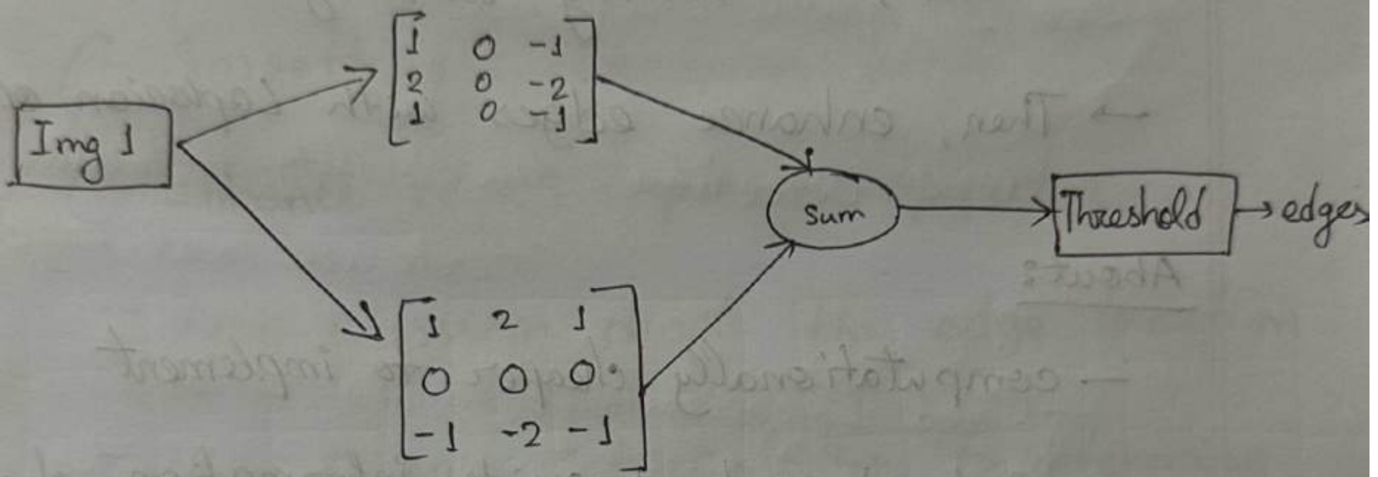
   ↳ take decision on the edge pixel where the changes are significant.

Extra 4. Localization

   ↳ determine the exact location of an edge.

Gradient based edge Detection:

By sobel,



- simple to implement
- capable of detecting edges and their direction
- sensitive to noise
- not accurate in locating edges.

Problems:

- too much details
  - ↳ one way to overcome this is to smooth images prior to edge ditection.
    - ↳ LOG

LOG (Laplacian of Gaussian):

→ First, smoothing with gaussian filter

→ Then, enhance edges with Laplacian operator

About:

– computationally cheaper to implement

– But doesn't provide information about the direction of the edge.

– Also probability of false and missing edges remain

→ Localization is better than Gradient operators.

LOC for Marr Hildreth Algorithm:

Steps:

- Smoothing: Gaussian filter
- Enhance edges: Laplacian operator
- Zero crossings denote the edge location
- Use linear interpolation to determine the sub-pixel location of the edge.

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

The Canny Edge Detector:

3 main criteria:

- Good Detection: the ability to locate and mark all real edges.

- Good Localization: minimal distance between the detected edge and real edge.

- Clear Response: only one response per edge.

Algo runs in 5 steps:

1. Smoothing
2. Finding gradients
3. Non-maximum suppression
4. Double thresholding
5. Edge tracking by hysteresis

এর কাজ strong edge গুলোকে খুঁজে নিয়ে আসা

Step 1: (smoothing / noise reduction).

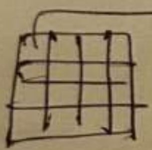
$$f_s(x, y) = G(x, y) * f(x, y)$$

$f_s(x,y)$ → filtered img

$G(x,y)$ → gaussian filter

$f(x,y)$ → original img

Step 2: (finding gradiant operator).



→ G_x & G_y calculate → $\sqrt{G_x^2 + G_y^2} = M(x, y)$

Step 3: (Non-maximum suppression)

$2\uparrow$	$3\uparrow$	$5\uparrow$	$4\uparrow$	$6\uparrow$
$4\uparrow$	$5\uparrow$	$7\uparrow$	$6\uparrow$	$7\uparrow$
$5\uparrow$	$6\uparrow$	$4\uparrow$	$3\uparrow$	$2\nearrow$
$3\uparrow$	$4\uparrow$	$3\uparrow$	$1\nearrow$	$1\nearrow$

Step 4: (Hysteresis Thresholding)

$$G_{NH}(x, y) = G_N(x, y) \geq T_H$$

$$G_{NL}(x, y) = G_N(x, y) \geq T_L$$

and

$$G_{NL}(x, y) = G_{NL}(x, y) - G_{NH}(x, y)$$

Step 5: (connectivity Analysis)

200	0	0
0	0	101
0	0	0

$g_{nH}(x, y)$

0	0	57
0	45	0
50	0	0

$g_{nL}(x, y)$

200	0	57
0	45	101
0	0	0

$g(x, y)$

☐ In Hysteresis thresholding, we set two thresholds 'High' and 'Low'.

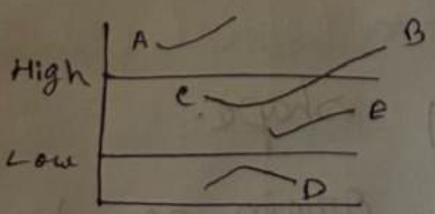
— If the threshold is set too Low, there will still be some false edge which is called False Positive.

— If the threshold is set too high, then actual valid edge points will be eliminated which is called False Negative.

=> High গুলার sure edge

=> Low গুলার non-edges

=> between 'High' and 'Low' হলে যদি High এর sure edge-এর সাথে connected থাকে then counted হয় otherwise discard করে দেয়া হয়। sure edge গুলার neighbor হলে connected বোঝায়।



Chapter - 9 (Morphological Image Processing)

Lecture - 11

- pixel shape based analysis
- basic morphology operations:
  1. Dialation → grow img regions
  2. Erosion → shrink img regions
  3. Opening → structured removal of img region boundary pixels.
  4. Closing → sturctured filling in of img region boundary pixels.

Binary Image:

representation 0, 1

→ foreground, object = 1 (white)

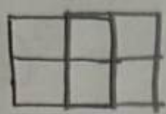
→ background = 0 (black)

Structuring Element:

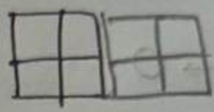
- can be any shape.
- has an origin. Origin can be in any position.

Fit:

Hit:



Miss:



Erosion:

Fit $\longrightarrow$ 1

Hit $\longrightarrow$ 0

Miss $\longrightarrow$ 0

$$A \ominus B = \{ Z \mid (B)_Z \subseteq A \}$$

Shrinks Foreground, enlarges Background

Erosion shrinks objects

Effect:

- Shrinks the size of foreground (1-valued) objects
- smoothes object boundaries
- Removes small objects

Dilation:

Fit → 1

Hit → 1

Miss → 0

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

$\boxed{\text{Enlarges foreground, shrinks background}}$

Dilation grows object

Effects:

- Fills in holes
- Smoothens object boundaries
- Adds an extra outer ring of pixels onto object boundary then object becomes slightly larger.
- Expands the size of foreground (1-valued) objects.

For Erosion,

$A - \{A \ominus B\}$ = Border (object এর inner line)

foreground pixel এর

For Dilation,

$\{A \oplus B\} - A$ = Border (object এর outer line)

background pixel এর

☐ Dialation and erosion are duals of each other :

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

Same way তে,

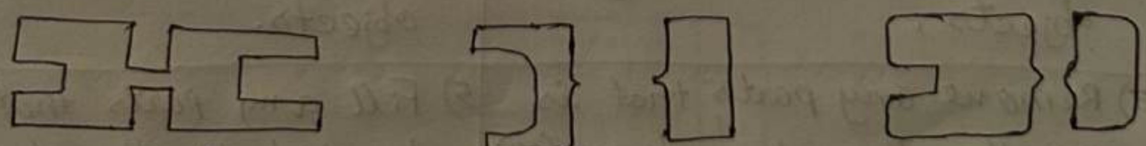
$$(A \oplus B)^c = A^c \ominus \hat{B}$$

Opening:

→ img এর উপর firstly erosion perform করবে।

→ Then সেই result আবার ঐটার উপর dilation perform করবে।

$$A \circ B = \{A \ominus B\} \oplus B$$



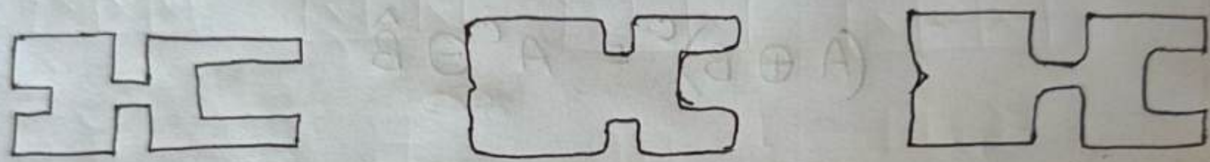
A $\qquad$ $A \ominus B$ $\qquad$ $\{A \ominus B\} \oplus B$

Closing:

→ img এর উপর firstly dilation perform করবো।

→ Then সেই result থেকে ওটার উপর erosion perform করবো।

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



A $\quad$ $A \oplus B$ $\quad$ $(A \oplus B) \ominus B$

Opening	Closing
1) Removes small objects/noise from the foreground of an img.	1) Fill small holes and gaps in the foreground of an img.
2) Erosion followed by dilation.	2) Dilation followed by erosion.
3) make objects smaller	3) Make objects larger
4) Useful for separating objects.	4) Useful for connecting objects.
5) Remove any parts that is smaller than the shape (SE) used.	5) Fill any parts that is touched by the shape (SE) used.
6) Denoted as: $A \circ B = (A \ominus B) \oplus B$	6) Denoted as: $A \bullet B = (A \oplus B) \ominus B$

Hit and Miss Transformation:

Lecture : 12

Hit and Miss Structure element contains :

0 , 1 and X (Don't care).

Here,

0 ⟶ background

1 ⟶ foreground

X ⟶ can be any pixel

WATCH OUT : We actually look for "fits" but we will be calling them "hits" when talking about Hit and Miss Transformation.

Structuring Elements representing 4 corners:

X	1	X
0	1	1
0	0	X

X	0	0
1	1	0
X	1	X

0	0	X
0	1	1
X	1	X

X	1	X
1	1	0
X	0	0

Equation:

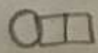
$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2)$$

Structuring Elements representing endpoints:

X	O	X
O	1	O
X	X	X

X	X	X
O	1	O
X	O	X

X	O	X
O	1	X
X	O	X



X	O	X
X	1	O
X	O	X