① (8) @ Noman I7 A Part :

Image Compression Models, why we need compress digital images:

$f(x,y) \rightarrow$ Source encoder $\rightarrow$ channel encoder $\rightarrow$ channel $\rightarrow$ channel decoder $\rightarrow$ Source decoder $\rightarrow f(x,y)$.

Image Compression is the Process of encoding or Converting an images file in such a way that is consumes less space than the original file. It is a type of Compression technique that reduces the size of an images file without affecting its quality to a greater extent.

Compressiong digital images is essential because it reduces their file size, making them easier to store and Process. High quality images can take up signiticant storage space, and compression helps minimize this without compression quality too much.
Smaller file size can save more images on our deivice and share them faster over the internet, even our connection slow. It also makes applications that uses images, website and editing software load and work more efficiency.
Reducing file sizes lowers storage and data transfer costs, especially for businesses using cloud services, Compression Can be losses, Preserving all images details, or lossy, which removes some details for smaller sizes, Both methods make digital images more Practical.

1(C) Write the JPG algorithm steps and Necessary Example.

(I) Color space Conversation: The image is converted RGB to YCbCr color space. Y represents brightness, while Cb and Cr carry color information. This helps focus more on brightness, which human eye is sensitive to.

(II) Down-sampling: The color information is reduced, as the human eye doesn't notice fine color's details. This saves space without significant quality loss.

(III) Divide image into 8x8 blocks: The image is divided into 8x8 pixels for easier Processing and efficient Compression.

(IV) Forward DCT: Each 8x8 block is transformed into the frequency domain, separating the image into components (edges, texture, flat regions).

(V) Quantization: The DCT coefficients divided by a quantization table and rounded. It removes small details to less noticeable and reducing file size.

(VI) Entropy Encoding: The quantized data is compressed using ~~[illegible]~~ methods huffman coding to further reduce the file size, creating ~~[illegible]~~ compressed final JPG file.

1(b): Encode and decode (4x4), $\boxed{8\text{ bits}}$ image using $\boxed{\text{LZW}}$ Coding.

208	208	129	129
208	208	129	129
208	208	129	129
208	208	129	129

প্রথমে ১টা, ২টা করে চলবে, যখন Previous মিলবে, তখন ৩টা, নতুন ২টা, তারপরেরটার করে, প্রতিটা একটা value (256 - 264) চলবে, ২টা হলে ১টা Dictionary থেকে encode output, তারপর location নাম্বার Entry থেকে নিবে,

Encode:

Dictionary Entry	Dictionary Location ~~Encoding~~	Encoding output
208	208	208
208 - 208	256	208
208 - 129	257	129
129 - 129	258	129
129 - 208	259	256
208 - 208 - 129	260	258
129 - 129 - 208	261	260
208 - 208 - 129 - 129	262	259
129 - 208 - 208	263	257
208 - 129 - $\boxed{129}$	265	129

output:

208 208 129 129
256 258 260 259
257 129

Decode এর encode থেকে নিবে, তারপর Dictionary মিলাবে, নতুন নাম্বার মিলবে, যদি হলে encode এটাই হবে থেকে value নিবে এর output

Decode:

Encode:	Dictionary Entry
	208
208	208
208	129
129	129
129	208 - 208
256	129 - ~~208~~ 129
258	208 - 208 - 129
260	129 - 208
259	208 - 129
257	129
129	

Decode output:

208 208 129 129
208 208 129 129
208 208 129 129
208 208 129 129.

LZW → encode and decode: 4x4 → 8bits:

200 200 129 129
200 200 129 129
200 200 129 129
200 200 129 129

0111010 111 1000 0 11011 1100 11010

01 → s
11010 → t
111 → a
10 → e
00 → i
11011 → o
1100 → u
11010 → t

a → 1x3
o → 5x1
t → 5x2
u → 4x1
i = 2x1
e = 2x1
s = 2x1

Encode:

Dictionary Entry	Dictionary Location	Encode output:
200	200	200
200 - 200	256	200
200 - 129	257	129
129 - 129	258	129
129 - 200	259	256
200 - 200 - 129	260	258
129 - 129 - 200	261	260
200 - 200 - 129 - 129	262	259
129 - 200 - 200	263	257
200 - 129 - $\boxed{129}$	264	129

output:

200 200 129 129
256 258 260 259
257 129

Decode:

Encode	Dictionary encode
200	200
200	200
129	129
129	129
256	200-200
258	129-129
260	200-200-129
259	129 - 200
257	200 - 129
129	129

Decode:

200 200 129 129
200 200 129 129
200 200 129 129
200 200 129 129

Data redundancy math: Huffman coding:

$a_2 \rightarrow 0.4$

$a_6 \rightarrow 0.3$

$a_1 \rightarrow 0.1$

$a_4 \rightarrow 0.1$

$a_3 \rightarrow 0.06$

$a_5 \rightarrow 0.04$

Symbol:	Probability	Code	1	2	3	4
a_2	0.4	1	0.4	0.4	0.4	0.6 0
a_6	0.3	00	0.3	0.3	0.3 00	0.4 1
a_4	0.1	011	0.1	0.2 010	0.3 01	
a_1	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

$$L_{avg} = \sum l(r_k) P_r(r_k)$$

$$= 0.4\times1 + 0.3\times2 + 0.1\times3 + 0.1\times4 + 0.06\times5 + 0.04\times5$$

$$= 2.2$$

$$C_R = \frac{n_1}{n_2} = \frac{3}{2.2} = 1.36$$

$$R_d = 1 - \frac{1}{C_R} = 1 - \frac{1}{1.36} = 27\%$$

Redundancy:

r_k	$P_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3		
$r_1 = 1/7$	0.25	001	3		
$r_2 = 2/7$	0.21	010	3		
$r_3 = 3/7$	0.16	011	3		
$r_4 = 4/7$	0.08	100	3		
$r_5 = 5/7$	0.06	101	3		
$r_6 = 6/7$	0.03	110	3		
$r_7 = 1$	0.02	111	3		

Symbol	Probability	code	1	2	3	4	5	6
r_0	0.19 →	11	0.19	0.19	0.19	0.19	0.19	0.81 0
r_1	0.25 →	01	0.25	0.25	0.25	0.25	0.56 00	0.59 1
r_2	0.21 →	10	0.21	0.21	0.21	0.35 10	0.25 01	
r_3	0.16 →	001	0.16	0.16	0.19	0.21		
r_4	0.08 →	0001	0.08	0.11	0.16			
r_5	0.06 →	00001	0.06	0.08				
r_6	0.03 →	000001	0.05					
r_7	0.02 →	000001						

$$L_{avg} = \sum l_2(r_k) \times P_r(r_k)$$
$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02)$$
$$= 2.7$$

$$C_R = \frac{3}{2.7} = 1.11,$$

$$R_d = 1 - \frac{1}{1.11} = 0.099 = 99\%$$

1(b): (i) Huffman coding generates Prefix-free minimum length binary code —
Huffman code ensures no code is a prefix of another, making decoding unambiguous.
Frequent characters get shorter codes, while less frequent ones get longer codes, minimizing the total bits. character: {a:5, b:9, c:12, d:13, e:16, f:45}
{f:0, c=100, d=101, a=1100, b=1101, e=111}

(ii) decode: 0111010111000110111001101 0

i=00, s=01, e=10, u=1100
t=11010, o=11011, a=111

01 → s, 11010 → t, 111=a, 10 → e, 00=i
11011 → o, 1100 → u, 11010 → t.

So, Decoding is → s→t→a→e→i→o→u→t.

(iii) Relative and Compression ratio → encoded using 8 bits.

01 → 2
11010 → 5
111 → 3
10 → 2
00 → 2
11011 → 05
1100 → 4
11010 → 5

Total Compression size = (2+5+3+2+2+5+4+5) = 28 bits.

original size = 8×8 = 64.

∴ Compression size $= \frac{64}{28} = 2.29$.

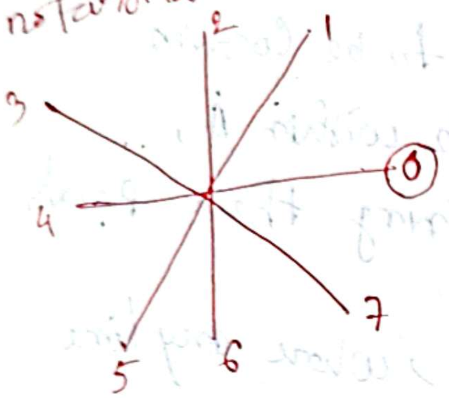
Data redundancy $= 1 - \frac{1}{2.29}$ ~~Compress/original~~
$= 1 - 0.43$
$= 0.57$
$= 57\%$.

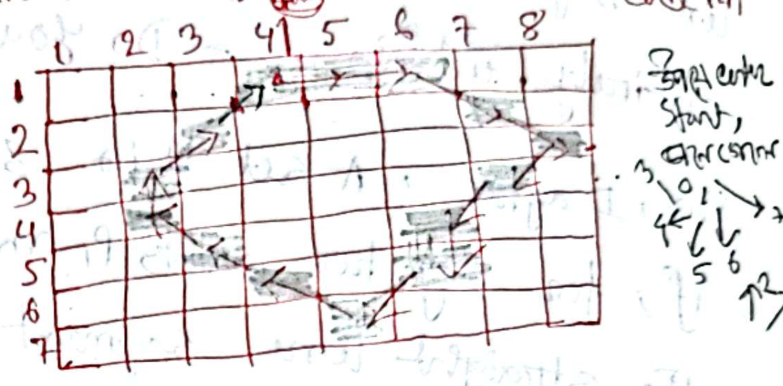
CamScanner

2 (a)

i) Represents the boundary figure -1 with an eighth (8) directional chain code.

ii) rotational invariant mean the case? make the chain code in notational invariant.





8 → chain code: 0 0 7 7 5 5 5 6 5 3 3 3 2 1 1

~~0 1 0 7 7 5 5 5 6 5 3 3 3~~

First diff → ~~0 1 0 7 7 5~~ 1 7 6 0 0 7 7 0

1 0 7 0 6 0 1 7 6 0 0 7 7 0

[illegible]

1 0 7 0 6 0 1 7 6 0

shape: 0 7 7 0 1 0 7 0 6 0 1 7 6 0

Preimeter = even count + $\sqrt{2}$ × odd count

$= 4 + 10 \times \sqrt{2}$

shape factor:

(ii) To make a chain code rotationally invariant,

The center point of the shape

Rotate the shape so that it points in a fixed directions, always starting from the top or along a specific axis.

Rotate the code so that the first direction always matches the fixed direction.

②(b): A is said to be convex if a straight line segment joining any points in A lies entirely within A - Do you agree, example →

Yes, I agree, A set A is said to be Convex if, for any two points P_1 and P_2 within A, the straight line segment joining these points lies entirely within A.

Convex: A circle is on square, where any line between two points stays inside.

Non-convex: A cresent L shape, where some lines go ~~to~~ outside the shape.

2(c): Find the ~~steps~~ on applying morphological Convex hull operations in an image.

extract the convex shape enclosing objects in a binary image.

① Load image: Read the input binary ~~to~~ image

② Threshold: If the image is grayscale, apply a threshold to convert it into a binary image.

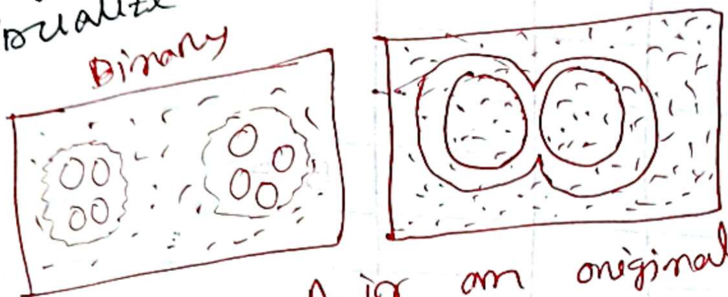
③ Identify Compometrs: Detect the individual Connected Components the binary image.

④ Apply convex hull: compute Convex hull using convex hull algo, and the ~~convex~~ smallest Convex polygon that encloses the objects.

(v) Draw the convex hull: Replace the original components with their convex hulls in binary images.
Merge all the convex hulls into a single binary image.
Visualize image with convex hulls applied.

Binary

2(a)://



Shown below A is an original binary image which has some defects. How would you detect the defects of the original image using morphological operations with different structural elements? Explain each structural element with proper example elements?

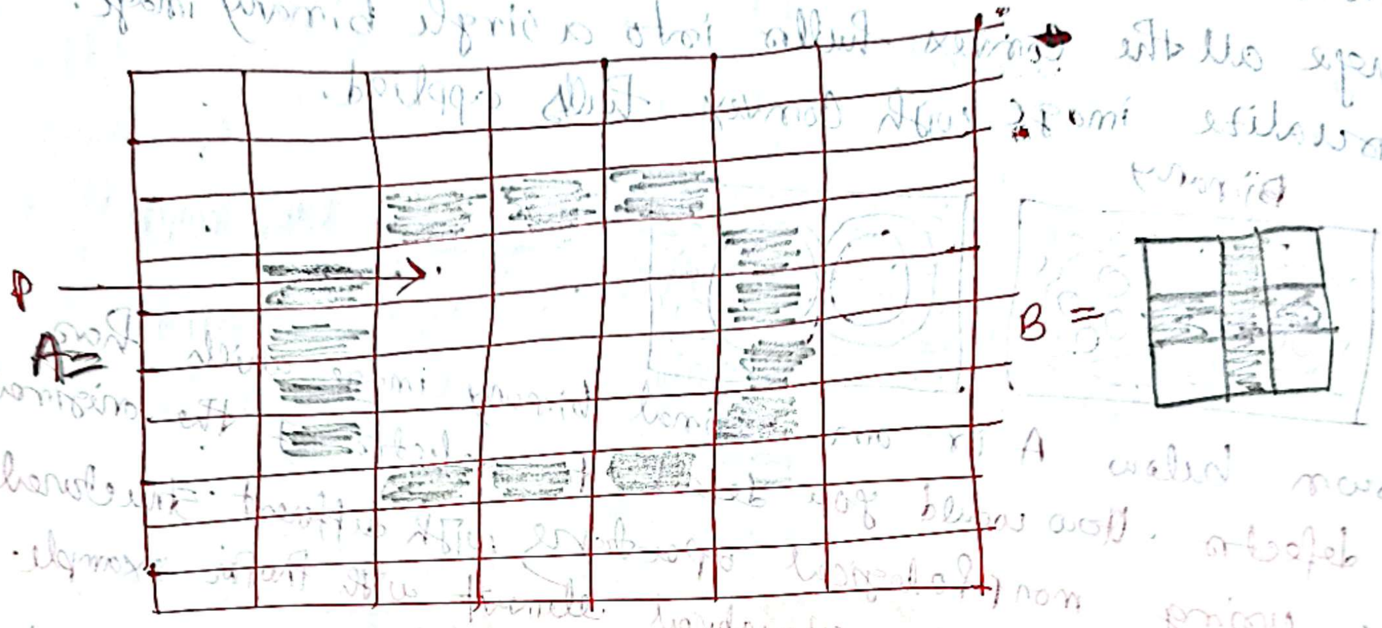
(i) Load images: load the binary image with objects as white and the background as black.

(ii) structural element: Detects general defects or noise.
Detect linear defects like scratches
Detects rounded or curved defects.

(iii) Morphological operations:
Expands white areas to fill gaps.
Shrinks white areas to remove noise.
Removes noises erosion by dilation.
Fills small holes.
Detects edges and boundary defects.

(iv) defects: defects = cv2.absdiff(original_image, processed_img)
The diff highlights the defects.

(b) Perform of region of filling Algorithm:



Explain:

⑧ Basically edge are created in an images due to the different types of discontinuity — ?

Yes, I agree. Edges in an images are created due to different types of discontinuity in Pixel intensity.

Intensity: when there is a sudden change in brightness or color between adjacent pixels, it form's an edge. A black object on a white background creates strong edges at the boundary.

~~Depth~~:

Depth: Changes in the depth of objects in a 3D scene create ~~so~~ edges when Projected into 2D images.

Surface orientation: Variations in surface angles cause intensity differences due to how lights reflects.

Texture: when there are abrupt changes in texture, edges appear even if the intensity remains constant.

⑪ Hit or miss → SE: 

Structuring elements: Define the SE with white 'T' shape Hit & a black square. The origin is usually the bottom left Corner of the black square.

Slide SE image: Move the SE over the binary image, one Pixel at a time.

Match Hit: check if the White 'T' in the SE aligns with the white pixels in the images.

Match miss: Ensure the black squre overlaps only black Pixels in the images.

Mark Match: Of both condition are met, Mark the location white in the output image otherwise Black. output → Highlists location SE Match exactly.

⑧ steps of applying convex hull operations.

Convex hull operations:

① Erosion (A⊖X) → shrink the white region in the image using structuring element (SE). smaller features may disappear if the SE is large.

② Complement (A′): Invert the eroded image. white becomes black, and black becomes white.

③ Dilation (W−X): Expands the boundaries of the white region using SE, fills small gaps or connects nearby regions.

④ Intersection (∩): Combines the result of ~~region~~ erosion and dilation. keeps only the areas that are white in both.

Final output: The resulting images highlights the convex hull or preserves the specific regions affected by the operations.

B Part:

3(a): In Edge linking similar points must be linked — what are the similar points — Give example.

Similar points refers to points along detected edges that share certain Properties and should be connected to complete continuous edge.

(i) Spatial Proximity → Points are considered similar if they are close to each other. In edge detection, small gaps between points can occur due to Noise detection.
If two edge points have coordinates (5,5), (5,6), they are spatially close & can be linkedup.

(ii) Gradient orientation → Points with similar orientations are likely belong to the same edge.
Point A at (10,20) & B(11,21) has a gradient direction of 45°. These points can be linkedup as they indicate same direction.

(iii) Intensity Similarity → The pixel intensities along the edge may also be compared. Points with similar intensity values might belong to the same edge.

Eg → After detecting edge points using the gradient, edge linked ensures disconnected segments are joined based on Proximity & direction.
→ for a circle in an image, if edge points along the curve are detected but have small gaps, linking fills in those gaps to complete the circle contour.

→ when detecting roads, small breaks in the detected edges are linked to from continous paths using similarity in direction & Proximity.

→ Write down the formulation of region in region oriented segmentation.

Region oriented segmentation divideds an image into regions based on specific criteria, such as intensity, texture, or homogeneity.

I represents image, R → Region,

$\bigcup_{i=1}^{n} R_i = I$ & $R_i \cap R_j = \phi$ for $i \neq j$.

A region R satisfied a uniformity condition

$f(R) = \text{Constant}$. $f(R) \to$ mean intensity.

$Var(R) \to$ variance intensity.

$\delta(R) \to$ The boundary of R should separate R from adjacent regions.

→ Segmentation on motion:

Involves identifying moving objects or regions in a video by analyzing motion patterns. It distinguishes the foreground from the background using motion detection, optical flow, background subtraction.

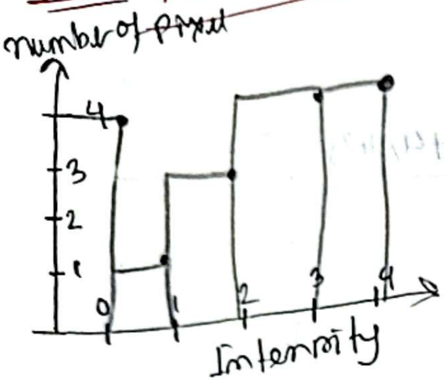
Frame diff: $M(x,y,t) = |I(x,y,t) - I(x,y,t-1)|$

optical flow : $\nabla I \cdot V + \frac{\delta I}{\delta t} = 0$.

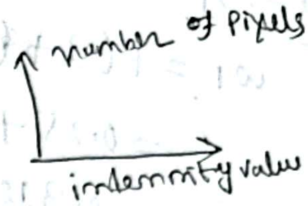
Application → Video surveillance, Autonomous vehicles, Activity recognition.

CamScanner

3(c): Find the optimal threshold using Otsu method:



0	1	3	0
4	2	3	2
4	4	3	4
0	2	0	3



variance, $\sigma_B^2 = W_b W_f (\mu_b - \mu_f)^2$

$W_{b,f}$ = Number of pixels in background / Total number of pixel

$\mu_{b,f}$ = Mean intensity of background.

weight w_f, Mean, μ_f, variance, σ_f^2

Intensity	frequency	$P_i = f_i/N$
0	4	0.25
1	1	0.0625
2	3	0.1875
3	4	0.25
4	4	0.25

$N = 16$.

$P_i = \frac{f(i)}{N}$

→ $w_1(T)$ → Background weight = $\sum_{i=0}^{T} P(i)$

$w_2(T)$ → foreground weight.

$= 1 - w_1(T)$.

→ class means;

Background mean; $\mu_1(T) = \frac{\sum_{i=0}^{T} i \cdot P(i)}{w_1(T)}$

Forehand mean; $\mu_2(T) = \frac{\sum_{i=T+1}^{4} i \cdot P(i)}{w_2(T)}$

Variance, $\sigma_b^2(T) = w_1(T) w_2(T) \cdot [\mu_1(T) - \mu_2(T)]^2$

→ Threshold, $T = 0, 1, 2, 3, 4$

① $\boxed{T = 0,}$ $w_1 = P(0) = 0.25$, $\mu_1 = \frac{0 \times 0.25}{0.25} = 0.00$

$w_2 = 1 - 0.25 = 0.75$, $\mu_2 = \frac{1 \times 0.0625 + 2 \times 0.1875 + 3 \times 0.25 + 4 \times 0.25}{0.75}$

$= 2.9167$

$\sigma^2(b)(T) = w_1(T) \cdot w_2(T) \cdot [(\mu_1(T) - \mu_2(T))^2]$

$= 0.25 \times 0.75 \times [(0 - 2.916)^2]$

$= 1.5951$.

$\boxed{T=1,}$

$w_1 = P(0) + P(1)$
$= 0.25 + 0.0625$
$= 0.3125$

$w_2 = 1 - 0.3125$
$= 0.6875$

$\mu_1 = \dfrac{0 \times 0.025 * 1 \times 0.0625}{0.3125}$
$= 0.2$

$\mu_2 = \dfrac{2 \times 0.1875 + 3 \times 0.25 + 4 \times 0.25}{0.6875}$
$= 3.0909$

$\sigma^2 (b)(T) = w_1(T) \cdot w_2(T) \cdot [(\mu_1(T) - \mu_2(T)]^2$
$= 0.3125 \times 0.6875 \cdot [(0.2 - 3.09)]^2$
$= 1.7955$

$\boxed{T = 2}$

$w_1 = P(0) + P(1) + P(2)$
$= 0.25 + 0.0625 + 0.1875$
$= 0.5$

$w_2 = 1 - w_1$
$= 1 - 0.5$
$= 0.5$

$\mu_1 = \dfrac{0 \times 0.25 + 1 \times 0.0625 + 2 \times 0.1875}{0.5}$
$= 0.875$

$\mu_2 = \dfrac{3 \times 0.25 + 4 \times 0.25}{0.5}$
$= 3.5$

$\sigma^2 = w_1(T) \cdot w_2(T) \cdot [\mu_1(T) - \mu_2(T)]^2$
$= 0.5 \times 0.5 \; [0.875 - 3.5)^2$
$= 1.7227.$

$\boxed{T=3}$

$w_1 = P(0) + P(1) + P(2) + P(3)$

$= 0.25 + 0.0625 + 0.1875 + 0.25$

$= 0.75$

$\mu_1 = \frac{0 \times 0.25 + 1 \times 0.0625 + 2 \times 0.1875 + 3 \times 0.25}{0.75}$

$= 1.5833$

$w_2 = 1 - w_1$

$= 1 - 0.75$

$= 0.25$

$\mu_2 = \frac{4 \times 0.25}{0.25}$

$= 4.0$

$\sigma^2 = w_1(T) \cdot w_2(T) \cdot [\mu_1(T) - \mu_2(T)]^2$

$= 0.75 \times 0.25 \cdot [1.583 - 4)^2$

$= 1.0951$

$\boxed{T = 4}$

$w_1 = P(0) + P(1) + P(2) + P(3) + P(4)$

$= 0.25 + 0.0625 + 0.1875 + 0.25$

$= 1$

$\mu_1 = \frac{0 \times 0.25 + 1 \times 0.625 + 2 \times 0.1875 + 3 \times 0.25 + 4 \times 0.25}{1}$

$= 2.1875$

$w_2 = 1 - 1$

$= 0$

$\mu_2 = 0.$

$\therefore \sigma^2 = 0.$

1st → কে কতবার আছে দিয়ে চার্ট. frequency বের করার, মধ্যমূল্য নিলাম: w_1, w_2, μ_1, μ_2 বের করার

T_0 $w_1 = P(0)$, তখন, $\mu_1 = \frac{P(0) \cdot (\text{intensity} \times \text{freq})}{w_1}$

$w_2 = 1 - w_1$ $\mu_2 = \frac{P(0) বাকি সবগুলো}{w_2}$

তারপর, $T_1 = w_1 = P(0) + P(1)$ একইভাবে বাকিগুলো, $\mu_1 = \mu_1$ এর মত, $\mu_2 = বাকি সবগুলো$

Optimal Threshold:

$T_1 = 1.7555$ for the Maximize the inter class variance and provides the best separation between background and foreground.

4(a): Describe your computing method of the skeleton of the images. What kind of structural elements will you be using? What happens when you changes the size of structuring elements.

To compute the skeleton of an images, a morphological thinning algorithm is applied. This process iteratively removes pixels from the boundaries of objects in the images while maintaining their connectivity and structure, reducing them to their central lines. A structuring element, typically a small matrix such as 3×3 or 5×5, is used to define the neighbourhood for thinning. The choice of structuring element influences the result, smaller elements preserve finer details, while larger simplify the skeleton by removing smaller features.

(4) What are the limitations of manual features engineering?

Manual features engineering is time consuming and requires significant domain knowledge to identify meaningful features. It can introduce human bias and may overlook important patterns in the data. This approach struggles with large, high-dimensional dataset and often fails to generalize well to new problems. It often misses complex relationships, limiting accuracy compared to automated methods.

~~[scribbled out]~~

4(d): ~~[scribbled out]~~ rotational invariant means in chain code? What does rotational invariant ... make the chain code rotational invariant? How can we

2-4+2 =-4

Rotational invariance in chain code means that the representation of a shape remains the same regardless of its orientation. To make a chain code rotationally invariant, the starting point of the code is chosen such that the sequence starts with the smallest numerical value when the code is treated as circular. This ensures the representation remains consistent, even if the shape is rotated.

4(c):

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

image

1	0	1
0	1	0
1	0	1

Filter

⇒ 1st - stride then

⇒ using max pooling.

For stride: TOP left, Top middle, Top right, Bottom ~~left~~, BOT-mid, BOT-right

Max pooling: Top left window, Top right window.

output height: (Img height - filter height) + 1

output width: (Image width - filter width) + 1

5 + 3 + 2 = 10

~~Top~~ (I) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \cancel{2+0}\ 4$

(II) $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 3$

(III) $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 4$

(IV) $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 2$

(V) $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 4$

(VI) $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 3$

(VII) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 2$

(VIII) $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 3$

(IX) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 4$

Convolution results:

$$\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Now, applying the max pooling,

2x2:

$\begin{bmatrix} 4 & 3 \\ 2 & 4 \end{bmatrix}$, max value = 4.

$\begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix}$, max value = 4

$\begin{bmatrix} 2 & 4 \\ 2 & 3 \end{bmatrix}$, max value = 4

$\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}$, Max value = 4

Final Max feature : $\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$

Vector : [4, 4, 4, 4].

This is the final result after convolution followed by max pooling.

⑧ 4(c):

Gray

6	3	6	3
4	4	2	4
2	2	1	3
6	1	3	2

Filter-1

-1	1	-2
-2	-3	2
2	2	3

Filter-2

-3	1	-2
-1	-3	1
2	3	3

Convolutional filter 3x3 gray. padding equal 2

2X2 max pooling.

Padding ~~images~~ is 2, the image will be padded with zeros on all sides. The padded images is 6x6 images.

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

-1	1	-2
-2	-3	2
2	2	3

Applying filter → 1:

① $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = 18$

(ii) $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 6 & 3 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = \cancel{16}\ \cancel{30}\ 21$

(iii) $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 6 & 3 & 6 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = 12$

(iv) ~~8~~ 1.5 (13) ~~26~~

(v) ~~8~~ 21 (14) 0

(vi) ~~20~~ 6 (15) -1

(vii) ~~22~~ 24 (16) 6

(8) 8 (17) -15

(9) 15 (18) -5

(10) 20

(11) 1

(12) 2

$$\begin{bmatrix} 18 & 21 & 12 & 15 & 21 & 6 \\ 24 & 8 & 15 & 20 & 1 & 2 \\ 26 & 0 & -1 & 6 & -15 & -5 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}$$

Now, again ~~fill~~ ~~pat~~ filter in 2nd,

$$\begin{bmatrix} 18 & 21 & 12 & 15 & 21 & 6 \\ 24 & 8 & 15 & 20 & 1 & 2 \\ 26 & 0 & -1 & 6 & -15 & -5 \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix} \begin{bmatrix} -3 & 1 & -2 \\ -1 & 3 & 1 \\ 2 & 3 & 3 \end{bmatrix}$$

(1)

Maxpooling. $\begin{bmatrix} 18 & 15 \\ 22 & 10 \end{bmatrix}$

$\begin{bmatrix} 18 & 15 & 22 & 10 \end{bmatrix}$

(5) Describe the application areas of Pattern recognition in brief:

Input
↓
Sensing
↓
segmentation
↓
Feature Extraction
↓
classification
↓
Post Processing
↓
Decision.

⇒ Pattern recognition is the science that concerns the description and classification of measurement of an object.

Input ⇒ The data we want to recognize, An image of a leaf.

sensing ⇒ Capture the data using a device, Taking a Photo of the leaf using camera.

segmentation ⇒ Divided the data into smaller, Parts for analysis, separating the leafs from the background in the Photo.

feature extraction: Identifies the important details, finding the colors ~~of the~~ spot on the leaf.

classification: Decided the categories or class of the data based on the features. Determine what are the leaf are healthy

Post Processing: Refines the result or adds additional steps for clarity.

Decision ⇒ The final output on conclusion based on the ~~output~~ analysis.

(iii) application areas of Pattern ~~Algiorss:~~ recognition:

definition: - - -

(i) Identify objects, faces or scenes in images for task ~~tit~~ facial recognition.

(ii) Converting spoken words into text for applications virtual assistant.

(iii) Detecting diseases using medical images, X-ray, CT scans, MRI,

(iv) Identifying patterns in stock trends or fraud detection in financial transactions.

(v) Interpreting handwriting text for applications postal services.

(vi) verifying identity using fingerprints, iris scans, voice patterns.

(iv) Recognization Algorithm → How can segment & classify object 8 bit images (4x4).

Image segmentation & classification are crucial steps in object recognition.

(i) sensing an image: This involves acquiring the 8-bit color image using camera or device.

(ii) Image segmentation: To isolate the specific object of interest from the background. If the object can be distinct color, simple threshold can be used, Canny edge detectors can identify boundaries between object & ~~boundari~~ background. watershed algorithm can group pixels with similar characterics.

Feature extraction: uses LBP.

- Divided the images into local regions, for each central pixels, compare its intensity with its neighbours. Created a binary ~~image~~ pattern for each region.

Classification uses ANN:

A powerful machine learning model inspired by the human brain. Collected the dataset of images with the object interest, corresponding LBP feature vector & class labels. The Network learns to map input features to the correct class. Present the features of a new images to the trained ANN. The Network predict the class.

※ Decision boundary ~~Vector~~ Pattern two class:
ω_1 & ω_2.
ω_1 & ω_2 where two mean vector → $M_1 = (4.3, 1.3)^T$
$M_2 = (1.5, 0.3)^T$.

Given, $M_1 = \begin{pmatrix} 4.3 \\ 1.3 \end{pmatrix}$

$M_2 = \begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix}$

$X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

&, $d_1(x) = xm_1 - \frac{1}{2} m_1 m_2$

The decisions functions —

for, $M_1 \rightarrow$ $d_1(x) = (x_1\ x_2) \begin{pmatrix} 4.3 \\ 1.3 \end{pmatrix} - \frac{1}{2} (4.3\ 1.3) \begin{pmatrix} 4.3 \\ 1.3 \end{pmatrix}$

$= 4.3x_1 + 1.3x_2 - \frac{1}{2}(18.49 + 1.69)$

$d_1(x) = 4.3x_1 + 1.3x_2 - 10.01$

$M_2 \rightarrow d_2(x) = (x_1\ x_2) \begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix} - \frac{1}{2} (1.5\ 0.3) \begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix}$

$= 1.5x_1 + 0.3x_2 - \frac{1}{2}(2.25 + 0.09)$

$= 1.5x_1 + 0.3x_2 - 1.17$.

$\therefore d_2(x) = 1.5x_1 + 0.3x_2$

The equation decision boundary —

$d_{12}(x) = d_1(x) - d_2(x)$

$= 4.3x_1 - 1.5x_1 + 1.3x_2 - 0.3x_2 - 10.1 + 1.17$

$= 2.8x_1 + 1.0x_2 - 8.9$ ...

$= 2.8x_1 + 1.0x_2 - 8.9$ ...

(I) Conv 2D (f = 16, 3x3, s = 1, P = 1)

Input shape = (256, 256, 3).

Filtersize = 3x3.

Stride = 1

Padding = 1.

output shape = (256, 256, 16)

(II) Maxpooling 2D (2x2, s = 2)

Input shape = (256, 256, 16)

poolsize = (2x2)

stride = 2

output = (128, 128, 16)

(III) Conv 2D (f = 32, 3x3, s = 1, P = 1)

Input shape = (128, 128, 16)

Filter size = 3x3.

Stride = 1.

Padding = 1

output = (128, 128, 32)

④ Max pooling 2D (2; 2; s = 2)

Input shape = (128, 128, 32)

pool size = 2×2

stride = 2

output shape = (64, 64, 32).

⑤ Convo 2D (f = 64, 3×3, s = 1, P = ?)

Padding : (3 − 1 / 2) = 1

Input = 64, 64, 32

Filter = 3×3

Stride = 1.

Padding = 1,

output = 64, 64, 64.

⑥ MaxPooling 2D (2×2, s = 2)

Input shape = 64, 64, 64

pool = 2×2

stride = 2.

output = 32, 32, 64.

~~Filter2D~~ ⑦. Flatten → Input = (32, 32, 64)

output = 65536. (32×32×64)

Decision boundary find ($M_1 < M_2$):

$5.1x_1 - 0.3x_2 - 8.43$ —— (i)

$6.6x_1 + 1.2x_2 - 28.33$ —— (ii)

$5.6x_1 + 3.8x_2 - 10.0$ —— (iii)

Decision boundary,

$5.1x_1 - 0.3x_2 - 8.43 = 0$ —— (i)

~~1st~~ Decision,

$6.6x_1 + 1.2x_2 - 28.33 = 0$ —— (ii)

2nd Decision,

$5.6x_1 + 3.8x_2 - 10.0 = 0$ —— (iii)

Now, equation (1)×4 + (2) →

$$20.4x_1 - 1.2x_2 = 113.32$$

$$6.6x_1 + 1.2x_2 = 28.33$$

$$33x_1 = 141.65$$

$$\therefore x_1 = \frac{141.65}{33} = 4.29$$

Now, in (ii) →

$$6.6 \times 4.29 + 1.2x_2 = 28.33$$

$$\Rightarrow 28.24 + 1.2x_2 = 28.33$$

$$\Rightarrow 1.2x_2 = 0.082$$

$$x_2 = \frac{0.082}{1.2} = 0.068$$

CamScanner

Now, (1) & (3) $\rightarrow$

$$5.1x_1 - 0.3x_2 = 8.43$$

$$5.6x_1 + 3.8x_2 = 10.01$$

$1 \times 3.8/0.3 \rightarrow$

$$64.6x_1 - 3.8x_2 = 106.78.$$

$$5.6x_1 + 3.8x_2 = 10.01$$

$$70.2x_1 = 116.79$$

$$x_1 = \frac{116.79}{70.2}$$

$$= 1.6$$

(iii) $\rightarrow$ $5.6x_1 + 3.8x_2 = 10.01$

$\Rightarrow 5.6 \times 1.6 + 3.82x_2 = 10.01$

$\Rightarrow 8.96 + 3.82x_2 = 10.01$

$\Rightarrow 3.82x_2 = 10.01 - 8.96$

$\therefore x_2 = 0.27.$

$M_1 = (4.28, 0.068)$

$M_2 = (1.6, 0.27).$

⑧ Write the edge linking algorithm?

In the edge linking algorithm, there are 3 steps,

(i) Apply edge detection algorithm such as Prewitt, sobel, robert, lapalacian, operator.

(ii) Calculate the similar point.

(iii) Link the similar point.

There are the three types of similarity —

(i) Similarity in magnitude.

(ii) Similarity in angle.

(iii) Both similar in magnitude & similar in angle.

⑧ what is the basic formula of region:

R

R_1	R_2	R_3
R_4	R_5	R_6
R_7		

(i) $\bigcup_{i=1}^{n} R_i = R$

(ii) R_i is a connected region, where $i = 1, 2, 3, \ldots n$

(iii) $R_i \cap R_j \neq \phi$ where $i \neq j$ & $i, j = 1, 2, 3, \ldots n$

(iv) $P(R_i) = $ true for $i = 1, 2, \ldots n$

(v) $P(R_i \cup R_j) = $ false for $i \neq j$ & $i, j = 1, 2, \ldots$

5041419 ...

(8) Write down the fundamentals steps in edge detection:

Edge detection is a key step in image processing to identify the boundaries of objects within an image.

Smoothing: Before detecting edges, we first reduce any noise or random variations in the image, which can make edges hard to identify. Noise could cause false edges that don't ~~bto~~ belong to real objects.

Finding gradients: This step finds the points of the image where the brightness changes most sharply. These changes indicate potential edges.

Thresholding: Thresholding helps decide which edges to keep by setting intensity limits. This ensures that only significant edges are kept, while weaker or less important ones are discarded.

Edge Tracking: In some methods, we add a final step to connect edges that might be slightly broken or interrupted. This helps create a continuous outline of objects.

Segment - 7

(8) Define Run Code, chain code & signature?

Run Codes

Run Code represents each row of an image by a sequence of length that describe successive runs of black and white pixel.

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

Runcode represents as (3,6, 5,4).

Chain Codes

chain codes are used to represents a boundary by a connected sequence of straight line segments of specified length and direction. Typically, this representation is based on 4 or 8 connectivity of the segments.

Signature:

A signature is a 1-D functional representation of a boundary and can be generated in various way.

(8) Disscuss the Different types of Artificial Neural Networks?

Two types of ANN:

(i) Feed forward Neural network: A feed forward Neural Network where connections between units do not form a directed cycle.

→ Starts at the Input layer, goes through hidden layer's and ends at the output layer.

It does not loop or revisit the same layer.

→ Predicting house Prices based on size, location, age.

(ii) Feed back Neural Network:

feedback Neural Network can have single travelling in the both directions by introducing loops in the Networks.

→ Starts like feedward, but sends some information back to adjust or improve the results. keeps memory of previous steps to make better Predications.

→ Predicting the next word in a sentence.

(8) Define Pattern, Pattern class, classifier, decision boundary.

(i) Pattern: A Pattern is a recognizable structure or set of features in data. It could be anything that repeats or has some specific structure. Patterns are what we try to identify or classify using a recognization system.

→ In an image, a face is a Pattern formed by features like eyes, nose & mouth.

A spoken word is a Pattern made up of sound waves with specific frequencies & rhythms.

(ii) A Pattern class is a group of similar patterns that share common characteristics or features. These are predefined categories into which Patterns can be classified.

→ If you have images of animals, "Cat" and "Dog" are two different Pattern class.

(iii) Classifier: A classifier is a tool, algorithm or model that assigns a Pattern to its corresponding Pattern class. It learns that the differentce between Patterns class and make decisions based on the features of the input data.

→ A classifier trained on animal images will look at features like fur, ears, tail to decide if an images of 'cat' or 'dog'

1) A decision boundary is the dividing line, curve or surface that separates different pattern ~~is~~ class in a feature space. It defines the rules for classification. Patterns on one side belong to one class, and those on the other side belong to another class.

→ For some complex data, the boundary might not be a straight line but a curve or irregular shape.

$\boxed{\text{Solve-Pw}}$

Math Practices:

Encode → Decode → LZW 4×4

$\boxed{208}$	208	129	129
208	208	129	129
208	208	129	129
208	208	129	129

For Encode →

Dictionary ~~start~~ entry	Dictionary Location	Encode output
208	$\boxed{208 \text{ initial}}$	
208 - 208	~~208~~ 256	208
208 - 129	~~208~~ 257	208
129 - 129	~~129~~ 258	129
129 - 208	~~129~~ 259	129
208 - 208 - 129	260	256
129 - 129 - 208	261	258
208 - 208 - 129 - 129	262	260
129 - 208 - 208	263	259
208 - 129 - 129	264	257
		129

208 208 129 129
256 258 260 259
257 129

Decode:

Encode	Dictionary entry	output Decode
208	208	208
208	208	
129	129	
129	129	
256	208 - 208	
258	129 - 129	
260	208 - 208 - 129	
259	129 - 208	
257	208 - 129	
129	129	

208 208 129 129
208 208 129 129
208 208 129 129
208 208 129 129

Huffman code:

Symbol	Probability	code	1 ~~code~~	2	3	4	5	6	code
r_0	0.19	00	0.19	0.19	0.19	0.19		0.85	
r_1	0.25	01	0.25	0.25	0.25	0.25	0.6	0.19	~~00~~
r_2	0.21	10	0.21	0.21	0.21	0.35	0.25		
r_3	0.16	001	0.16	0.16	0.19	0.25			
r_4	0.08	0001	0.08	0.11	0.16				
r_5	0.06	00001	0.06	0.08					
r_6	0.03	000001	0.05						
r_7	0.02	000001							

Now, $L_{avg} = (0.19\times 2) + (0.25\times 2) + (0.21\times 2) + (0.16\times 3) + (0.06\times 5) + (0.03\times 6) + (0.02\times 6) + (0.08\times 4)$

$= 2.7$

Now $C_r = \dfrac{N_1}{N_2} = \dfrac{3}{2.7} = 1.1$

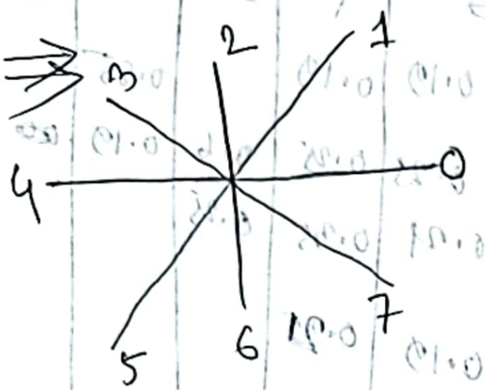
Now, $R_d = 1 - \dfrac{1}{C_r}$

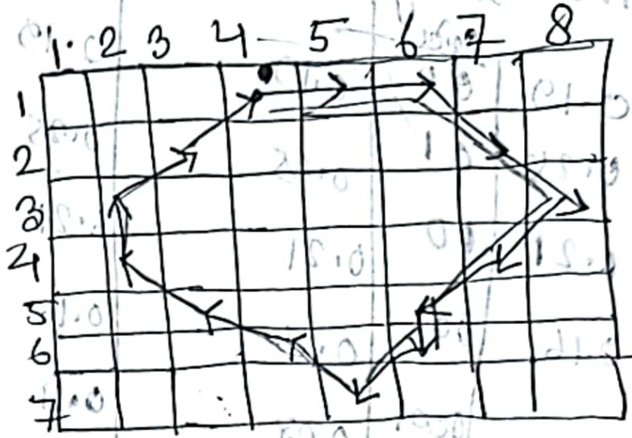
$= 1 - \dfrac{1}{1.1}$

$= 0.09$

$= 9\%$

Directed 8 chain code → math:





Chain code → 0 0 7 7 5 5 6 5 3 3 3 2 1 1

Rotational 45-shape →

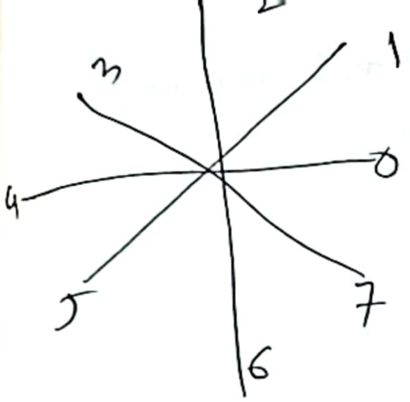
1 7 0 6 0 1 7 6 0 3 7 7 0

Perimeter = even count + $\sqrt{2}$ odd count

$= 4 + 10\sqrt{2}$

Now, Area:

Area	Code	Value of Y
Area = Area + y	0	y = y
Area = Area + $(y+\frac{1}{2})$	1	y = y + 1
Area = Area	2	y = y + 1
Area = Area − $(y+\frac{1}{2})$	3	y = y + 1
Area = Area − y	4	y = y
Area = Area − $(y-\frac{1}{2})$	5	y = y − 1
Area = Area	6	y = y − 1
Area = Area + $(y-\frac{1}{2})$	7	y = y − 1



$\underline{0}\,\underline{7}\ \underline{7}\ \underline{7}\,\underline{5}\ \underline{5}\ \underline{6}\ \underline{5}\ \underline{3}\ \underline{3}\ \underline{3}\ \underline{2}\,1\,1$

$$\Rightarrow y^{\to y} + y^{\to y} + \left(y-\tfrac{1}{2}\right)^{\to y-1} + (y-1.5)^{\to y-2} - (y-2.5)^{\to y-3}$$

$$- (y-3.5)^{\to y-4} + 0^{\to y-5} - (y-5.5)^{\to y-6}$$

$$- (y-5.5)^{\to y-5} - (y-4.5)^{\to y-4}$$

$$- (y-3.5)^{y-3} + 0^{\to y-2} + \left(y-\tfrac{1}{2}\right)^{\to y-1}$$

$$+ (y-.5)^{\to y}.$$

$$= y + y + y - 0.5 + y - 1.5 - y + 2.5$$

$$- y + 3.5 + 0 - y + 5.5 - y + 5.5$$

$$- y + 4.5 - y + 3.5 + 0 + y - 0.5$$

$$+ y - 0.5$$

$$= 6y - 6y + 22$$

Area $= 22$

$\therefore$ Shape factor $= (\text{Perimeter})^2 / \text{Area}$

$$= \frac{(4+10\sqrt{2})^2}{22}$$

$$= 2.3377$$

Rotationally, The center point of the shape $\left(\frac{M}{2}\right)$. Rotate the shape so, that it points in a fixed directions, always starting from the top or along a specific axis.

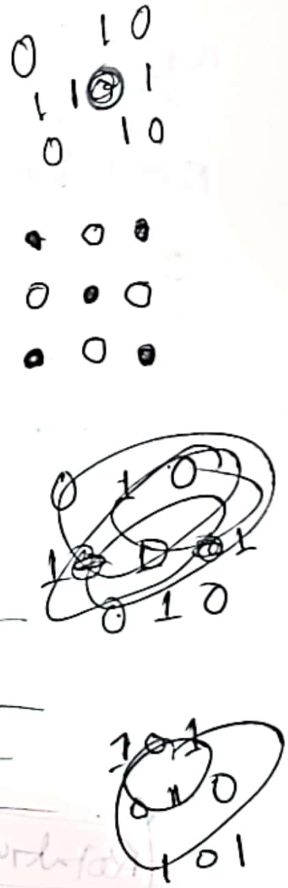
Filling Algorithm:

A be the set of 8 connected boundary points of a region, Start with a point inside the region dilate,

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

errioson: $\boxed{A \ominus B}$ → structure element full match হলে origin ① থাকবে নয়তো 0.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
0	0	0	1	1	1	1	0	1	0	0	1	1	1	[illegible]	0
0	0	1	1	1	1	0	1	1	1	1	1	01	0	0	0
0	1	1	1	1	0	0	0	1		0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0
0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0
0.	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0
0.	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



⇒ Erosion condition, ① check full structuring image element that match in A.

Match do again 1 except all 0.

Remove pixel.

Decrease brightness!

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

Otsu method : Threshold

0	1	3	0
4	2	3	2
4	4	3	4
0	2	0	3

Now,

Intensity	frequency	$P_i = f_i/N$
0	4	$4/16 = 0.25$
1	1	$1/16 = 0.0625$
2	3	$3/16 = 0.1875$
3	4	$4/16 = 0.25$
4	4	$4/16 = 0.25$

$N = 16$

Threshold, $T = 0, 1, 2, 3, 4.$

Now, $W_1(T) = \sum_{T_i=0} P(i)$

$W_2(T) = 1 - W_1$

$$\mu_1 = \frac{\sum_{i=0} i \cdot P(i)}{W_1(T)}$$

$$\mu_2 = \frac{\sum_{i=T+1}^{4} i \cdot P(i)}{W_2(T)}$$

Variance, $\sigma_B^2(T) = W_1(T) \cdot W_2(T) \times [\mu_1(T) - \mu_2(T)]^2$

Now, for, $\boxed{T(0) = 0}$,

$W_1 = P(0) = 0.25$.

$W_2 = 1 - W_1$

$= 1 - 0.25$

$= 0.75$.

$\mu_1 = \dfrac{0 \times 0.25}{0.25}$

$\mu_2 = \dfrac{1 \times 0.0625 + 2 \times 0.1875 + 3 \times 0.25 + 4 \times 0.25}{0.75}$

$= 2.9167$

Variance $\sigma^2(b)(T) = 0 \times 0.75 * (0 - 2.91)^2$

$= 1.593$.

$\boxed{T(0) = 1}$,

$W_1 = P(0) + P(1)$

$= 0.25 + 0.0625$

$= 0.3125$

$W_2 = 1 - 0.3125 = 0.6875$

$\mu_1 = \dfrac{0 \times 0.25 + 1 \times 0.0625}{0.3125} = 0.2$

$\mu_2 = \dfrac{2 \times 0.1875 + 3 \times 0.25 + 4 \times 0.25}{0.6875}$

$= 3.0909$

$\sigma^2(b)(T) = 0.3125 \times 0.6875 [0.2 - 0.6875]^2$

$= 1.7955$

(8)

$$\begin{bmatrix} 6 & 3 & 6 & 3 \\ 4 & 4 & 2 & 4 \\ 2 & 2 & 1 & 3 \\ 6 & 1 & 3 & 2 \end{bmatrix} \qquad \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$

(i) $\begin{bmatrix} 6 & 3 & 6 \\ 4 & 4 & 2 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = 0 - 4 + 3$

$= -1$

(ii) $\begin{bmatrix} 3 & 6 & 3 \\ 4 & 2 & 4 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = -9 + 6 + 7$

$= 4$

(iii) $\begin{bmatrix} 4 & 4 & 2 \\ 2 & 2 & 1 \\ 6 & 1 & 3 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = -8 - 2$

$= -11$

(iv) $\begin{bmatrix} 4 & 2 & 4 \\ 2 & 1 & 3 \\ 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} -1 & 1 & -2 \\ -2 & -3 & 2 \\ 2 & 2 & 3 \end{bmatrix} = 0 + 5 + 10$

$= 15$

matrix $= \begin{bmatrix} -1 & 4 \\ -11 & 15 \end{bmatrix}$

maxpooling $= 15$.

3) Mean two vector → w_1 & w_2

$$m_1 = [6.1, 1.1, 5.0]$$

$$m_2 = [2.2, 0.2, 2.4]$$

$$x = [2.1, 0.4, 3.5].$$

$$g_1(x) = -\frac{1}{2}\left((x-m_1)^T (x-m_1)\right)$$

$$g_2(x) = -\frac{1}{2}\left((x-m_2)^T \cdot (x-m_2)\right)$$

$$d_1 = \Sigma(x-m_1)^2$$

$$= \left((2.1-6.1)^2 + (0.4-1.1)^2 + (2.4-5.0)^2\right)$$

$$= 16 + 0.49 + 2.25$$

$$= 18.74.$$

$$d_2 = \Sigma(x-m_2)^2$$

$$= \left((2.1-2.2)^2 + (0.4-0.2)^2 + (3.5-5.0)^2\right)$$

$$= 0.01 + 0.04 + 1.21 = 1.26.$$

$$g_1(x) = -\frac{1}{2} \times 18.74$$

$$= -9.37$$

$$g_2(x) = -\frac{1}{2} \times 1.26$$

$$= -0.63$$

$M_1 = (4.3, 1.3)^T$

$m_2 = (1.5, 0.3)^T$

$M_1 = \begin{pmatrix} 4.3 \\ 1.3 \end{pmatrix}$

$m_2 = \begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix}$

$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$d_1(x) = x m_1 - \frac{1}{2} m_1 m_2$

Now from,

$m_1 = d_1(x) = (x_1 \; x_2)\begin{pmatrix} 4.3 \\ 1.5 \end{pmatrix} - \frac{1}{2}(4.3 \; 1.3)\begin{pmatrix} 4.3 \\ 1.3 \end{pmatrix}$

$= 4.3x_1 + 1.5x_2 - \cancel{8.6} \; \frac{1}{2}(18.49 + 1.69)$

$d_1(x) = 4.3x_1 + 1.5x_2 - 10.09$

$m_2 = d_2(x) = (x_1 \; x_2)\begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix} - \frac{1}{2}(1.5 \; 0.3)\begin{pmatrix} 1.5 \\ 0.3 \end{pmatrix}$

$= 1.5x_1 + 0.3x_2 - 1.57$

boundary, $d_{12}(x) = d_1(x) - d_2(x)$

$= 2.8x_1 + 1.2x_2 - 11.58$

CNN :

(i) Convo 2D (f=16, 3×3, s=1, p=1)

Inputshape : 256, 256, 3

filter : 3 × 3

Stride : 1.

padding : 1.

output : 256, 256, 16.

(ii) maxpooling (2×2, s=2).

Input shape : 256, 256, 16

~~F~~ilter : 2×2

Stride : 2.

output : 128, 128, 16

(iii) Convo2D (f=32, 3×3, s=1, p=1)

Input shape : 128, 128, 16

~~f~~ilter : 3×3

stride : 1.

output shape : 128, 128, 32.

(iv) maxpooling2D (2×2, s=2)

Inputshape : 128, 128, 32

~~f~~ilter : 2×2

stride = 2.

output : 64, 64, 32

⊖ Conv (f = 64, 3×3, s = 1, p = ?)

$p = \frac{3-1}{2} = 1$

Pixel : 1.

filter : 3×3

Stride = 1.

output = 64, 64, 64.

⑨ maxpool (2×2, s = 2)

Input shape : 64, 64, 64

filter : 2×2

Stride : 2

outputshape : 32, 32, 64.

flatten output : (32×32×64)

= 65536.

$m_1 = [6.1, 1.1, 5]$

$m_2 = [2.2, 0.2, 2.4]$.

$x = [2.1, 0.4, 3.5]$

$g_1(x) = -\frac{1}{2}((x - m_1)^T (x - m_1)$

$g_2(x) = -\frac{1}{2}((x - m_2)^T (x - m_1)$

$d_1(x) = \Sigma(x - m_1)^2$

$= (2.1 - 6.1)^2 + (0.4 - 1.1)^2 + (3.5 - 5)^2 = 18.74$

$d_2(x) = \Sigma(x - m_2)^2$

$= (2.1 - 2.2)^2 + (0.4 - 0.2)^2 + (3.5 - 2.4)^2$

$= \cancel{\ } \; 1,26$

$\therefore g_1(x) = -\frac{1}{2} \times 18.74 = -9.32$

$\therefore g_2(x) = -\frac{1}{2} \times 1.26 = -0.63$.