

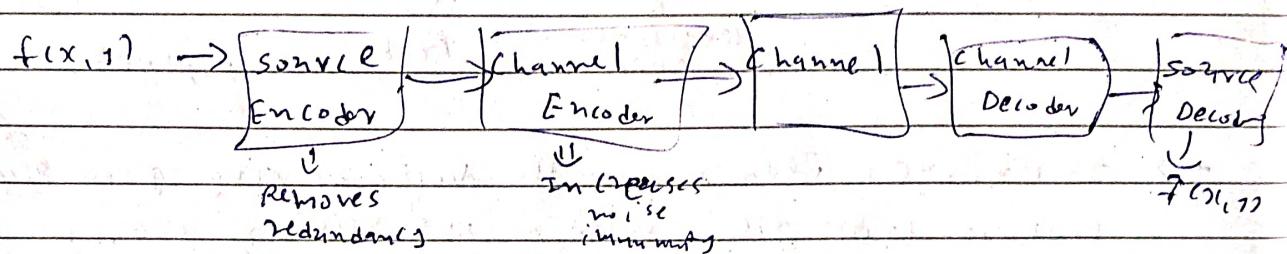
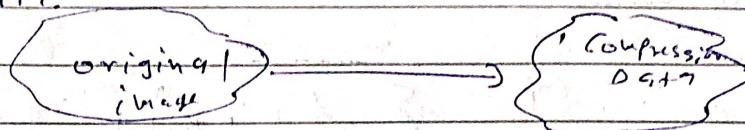
A Image compression

→ Image compression refers to the process of reducing the quantity of data required to represent the given quantity of information for digital image. The basis of reduction process is removal of redundant data.

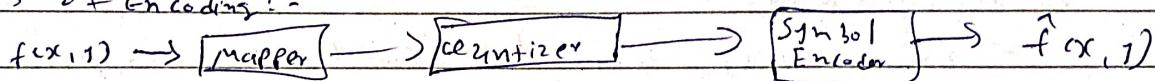
methods

(1) Huffman coding

(2) Arithmetic & transform coding



Stages of Encoding:-

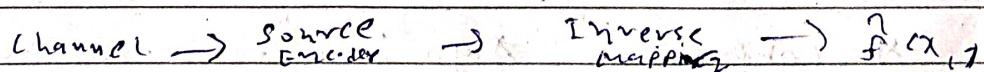


Mapper → Reduces interpixel redundancy, reversible operation

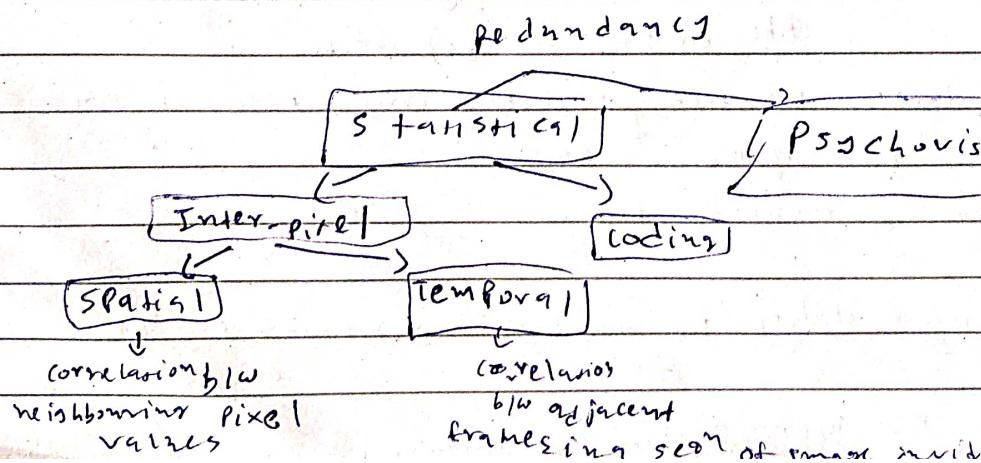
Quantizer → Reduces physiovisual redundancy, non-reversible op.

Symbol Encoder → Create a fixed or variable length code, Reversible

Source Decoder:-



A Redundancy: Redundancy means Repetitive data



→ Coding Redundancy : It happens due to poor selection of coding technique

Coding Redundancy = Avg. bits used to code - Entropy

$$R = \sum_{k=0}^n l(r_k) P(r_k) - \sum_i p_i \log_2 \left(\frac{1}{p_i} \right)$$

length of code ↓ Prob. of fragment

→ InterPixel Redundancy : The value of any given pixel can be predicted from the value of its neighbors that is they are highly correlated.

→ It is solved by algs. like run length coding, Bit plan algo.

→ Psycho-visual Redundancy

- It exists because human perception does not involve quantitative analysis of every pixel in the image
- loss of information

* Compression Algo : It reduce the source data to a compressed form and decompress it to get original (image) data.

→ Two types of CA

(1) Lossless compression : Reconstructed data is identical to the original (image) data.

- It means reducing the size of data in image without any quality loss. ex. Huffman, Arithmetic
- < remove unnecessary metadata from JPEG & PNG

(2) Lossy :

- Reconstructed data is an approximation to the original data
- ex. linear predict of transform code

diff

lossless compress.

- file can be restored in its original
- quality doesn't compromised
- It does not reduce the size
- ex. Huffman & Runlength
- It used in text, image, speech
- does not eliminate

lossy compress.

- Not restored
- compromised
- reduce the size of data
- ex. Transform coding & wavelet
- It used in images, audio, video
- It eliminates data which is not noticeable

• Huffman Coding :- It reduces an avg. no. of bits / pixel.

- It assigns variable length bits to different symbols
- Achieve compression in 2 steps.

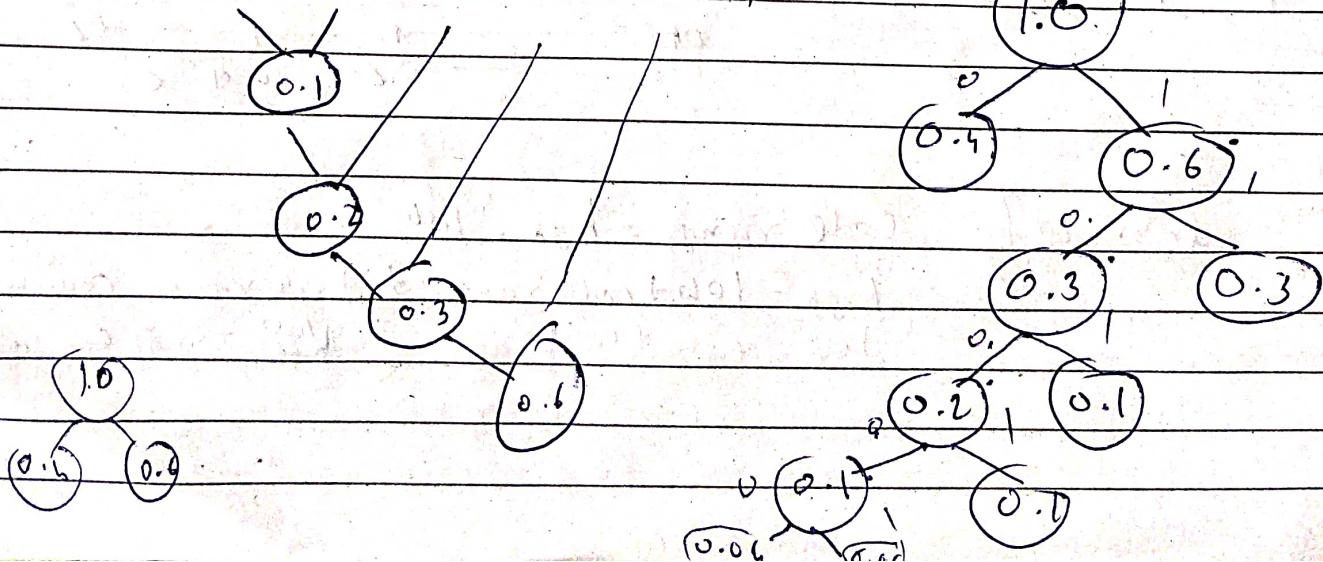
1) Source reduction

2) Code Assignment

Ex.

State	P				
a ₁	0.4	0	0.4	0.4	
a ₂	0.3	11	0.3	0.3	0.6
a ₃	0.1	101	0.1	0.1	0.2
a ₄	0.1	1001	0.1	0.1	0.1
a ₅	0.06	10000	0.1	0.2	0.6
a ₆	0.04	10001		0.4	1

(1) encode a₆ as a₄ as a₃ a₂ a₁.



$$L = \text{Avg. Length of code} = \sum p_k l_k$$

$$\begin{aligned} \text{Length of code} &= 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.06 \times 5 \\ &= \text{ans} \end{aligned}$$

$$H(x) = \text{Entropy} = \sum_{k=1}^n p_k \log_2 \frac{1}{p_k} = 0.4 \times \log_2 \frac{1}{0.4} + 0.3 \times \log_2 \frac{1}{0.3} + \dots + 0.06 \times \log_2 \frac{1}{0.06}$$

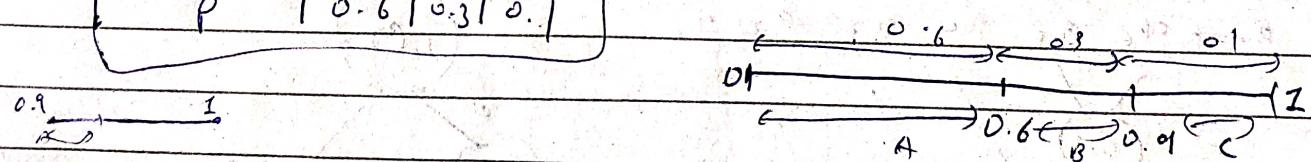
$$\text{Coefficient } n = H = \frac{100}{L} \rightarrow \text{ans in } \%$$

Arithmetic Coding: It is an alternative to Huffman compression, it enables characters to be represented as fractional bit lengths.

e.g. Code CA B using arithmetic coding.

Character	A	B	C
P	0.6	0.3	0.1

(1) Divide the ranges into 0-1



$$A = 0.9 + 0.1 \times 0.6 =$$

$$B = 0.9 + 0.1 \times 0.3 =$$

$$C = 0.9 + 0.1 \times 0.1 =$$

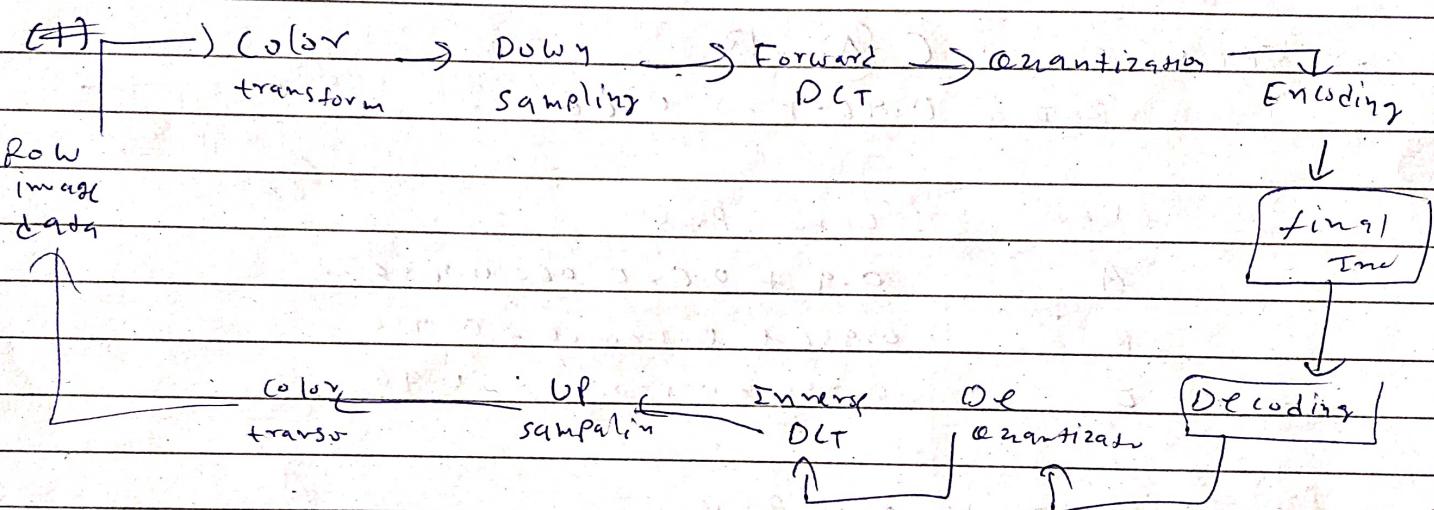
formula

Code range = high - low

high = low + code range * high range symbol

low = (low + code range * high range symbol)

- * JPEG/JF / frequency based compression
- It is a commonly used method of lossy compression for digital image.
 - It uses discrete cosine transformation coding (DCT)
 - It allows a tradeoff b/w storage size and the degree of compression can be adjusted.
 - It is followed by 8 steps.



Step 1 : Input image split into 8x8 pixel blocks

Step 2 : It uses Y, Cb, Cr Model so convert RGB to YCbCr

Step 3 : It is forwarded to DCT. DCT uses cosine of Y to convert into frequency domain

$$f(u, v) = \frac{c_3}{2} \frac{c_v}{2} \sum_{x=0}^2 \sum_{y=0}^2 f(x, y) \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

$$c_0 = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u=0 \\ 1, & \text{if } u>1 \end{cases}$$

$$f(x, y) = \sum_{u=0}^2 \sum_{v=0}^2 f(u, v) c_u \frac{c_v}{2} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

split $\xrightarrow{\text{why}} \text{DCT} \xrightarrow{\text{co-anti}}$ zigzag $\xrightarrow{\text{vectorize}}$ RLE $\xrightarrow{\text{Huffman}}$

Step 5: quantization is used to reduce the no. of bits per sample.

$$f(3, i) = \text{round} \left(\frac{f(3, v)}{Q(3, v)} \right)$$

Step 5: The zigzag scan is used to scan the 8x8 matrix to 1x64 vector. It is used to group 100 freq. coefficients to top level of vector & high freq to the bottom.

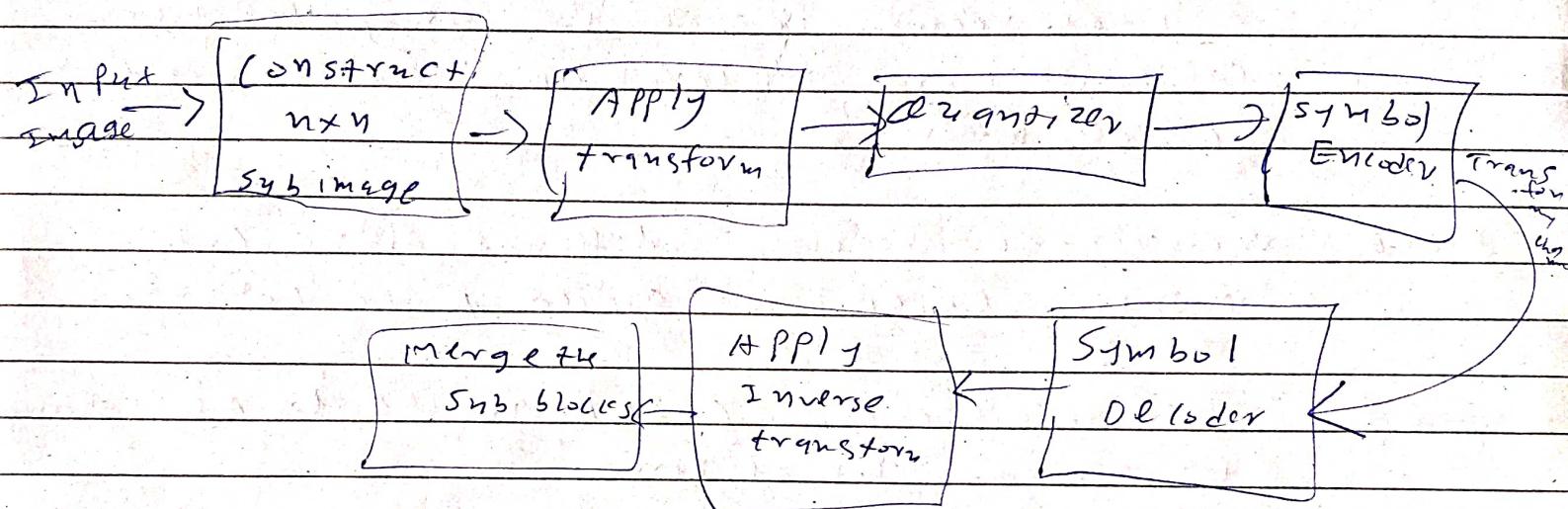
Step 6: Vectorizing is used for different Pulse Code Modulation (PCM) is applied to DC component.

Step 7: Run length coding (RLE) is applied to AC components for removing lot of zeros.

Step 8: Finally apply Huffman coding to the ~~DC~~ and get our

BLOCK Transform Coding

- Transform coding compresses image data by representing the original signal with a small number of transform coefficients.



(1) → Sub image selection:

- At this stage image is divided into a set of sub images.
- The values of n is a power of 2
- This step is necessary to reduce computation complexity.

(2) → transform selection

- Idea behind transform coding is to use mathematical transforms for data compression.
- Transformations like DFT, DCT and Wavelet transforms are used.
- Choices of transforms depends on the resources and amount of error associated with reconstruction process.

7w b20

7c 2 L 18

→ Bit Allocation

- Transform Coding is the process of quantizing, allocating and coding the coefficients of the transformed sub image.
- minimum distortion

→ Zonal Coding

- Multiplying each transform coefficient by the corresponding element in zonal mask

→ threshold mask

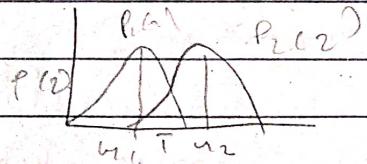
- Threshold coding works based on the fact that transform coefficients having maximum magnitude make the most contribution to the image

* Thresholding : // Image segmentation //

- Thresholding plays a very important role in segmentation
- This makes a difference between the object & background of image

* Thresholding may be defined as

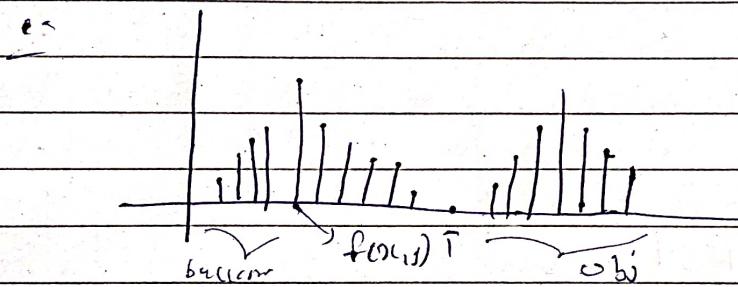
- single level thresholding
- multi-level thresholding



$$P_{(2)} = P_1 p_{1,(2)} + P_2 p_{2,(2)}$$
$$P_{(2)} = \sum_i P_i p_{i,(2)}$$

* Single level :

- Suppose that an image $f(x, y)$, has light objects on a dark background. If we draw the histogram of this type of image, it will have two dominant modes of gray levels.



- one group near the zero point of histogram showing dark background and one group near the maximum point of histo showing light image

so, at any point (x, y) , if

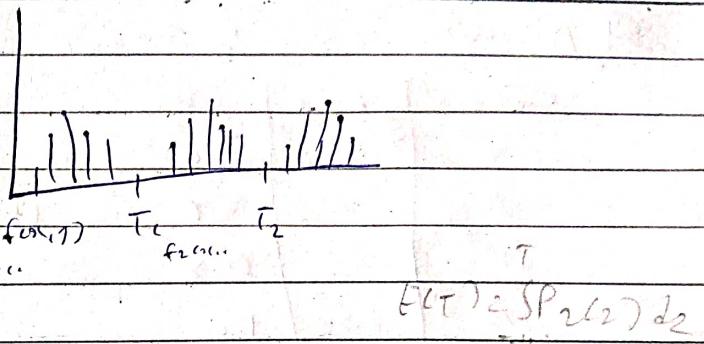
$f(x, y) > T$, it is called object

$f(x, y) < T$, it is called background

→ So it called b/c we have separated info of an image by single threshold

(2) Multilevel Thresholding.

→ More than one thresholding.



* Types of thresholding

1) Global Thresholding: If thresholding depends upon only the gray level of the image.

2) Local Thresholding: If thresholding depends upon the gray level of image and some local property.

3) Adaptive thresholding: When thresholding depends upon gray level, local properties and spatial co-ordinates of pixels. It is also called dynamic thresholding.

* There is if filter mask are given of any image then we can find the threshold value by using that formula

$$\text{Threshold} = \frac{1}{2} (m_1 + m_2)$$

→ where m_1 and m_2 are the avg of pixels

→ we can devide in two group one is m_1 is greater than avg pixel value (take avg of greater than the avg values)

→ m_2 is less than the avg pixel value & take the avg.

→ then put in formula

Ex.

4	2	3
8	9	1
2	5	1

$$m_1 = \frac{4+2+3+8+9+1+4+5+1}{9} = \text{avg}$$

$$\text{avg} = \frac{\text{sum}}{\text{total no.}}$$

$$m_2 = \frac{\text{sum}}{\text{total no.}}$$

Threshold = $\frac{m_1 + m_2}{2}$

- optimization
 - convexity
 - uniqueness
 - off
- CMIN
→ ANIV

* Thresholding

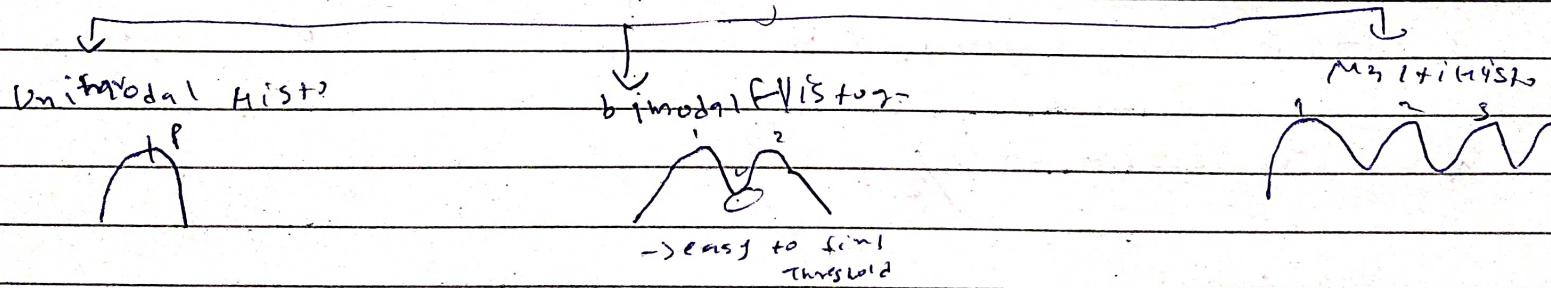
- It is an important technique for image segmentation
- Key It produces uniform regions based on the threshold criteria
- Key parameter of thresholding process is the choice of threshold value
- Global Thre → depends upon only grey scale value
- Local Thre → depend on grey scale and some neighborhood property
- Dynamic Thre → It case it depends on Pixel coordinates also
[grey + property + coordinates]

$$Th = T[x, y, f(x, y), A(x, y)]$$

* Histogram and Thresholding.

- Tool that help to find the threshold hist.

Types of histo

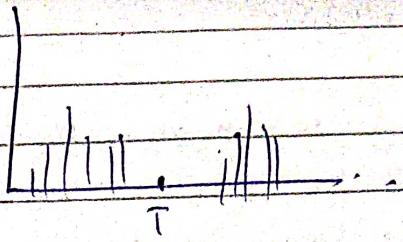


∴ we can select the random value as threshold in some case

→ suppose the an image $f(x, y)$ is composed of light objects on the dark background and shown in fig.

Simple level threshold

→ The objects can be extracted by comparing pixel values with the threshold T

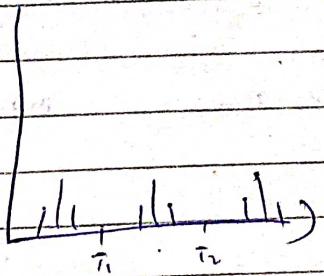


at any point if $\begin{cases} f(x,y) > T & \text{it is an object image} \\ f(x,y) < T & \text{it is background image} \end{cases}$

• Multilevel threshold

if $T_1 < f(x,y) \leq T_2 \rightarrow$ one object class
 $f(x,y) > T_2 \rightarrow$ other object class

def: it is also possible to extract objects that have specific intensity range using multiple thresholds



• Global Threshold:

→ histo have two distinct separated by a ~~value~~ between them

$$f(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{otherwise} \end{cases}$$

→ $T_{\text{max}} = (m_1 + m_2) / 2$

• Multilevel threshold

→ multilevel thresholding is the extension of simple thresholding tech.
 $f_i = \text{Value of input image pixel}$

$t_1, t_2, \dots, t_n \rightarrow$ multiple threshold values

$s_1, s_2, \dots, s_n \rightarrow$ value of output image

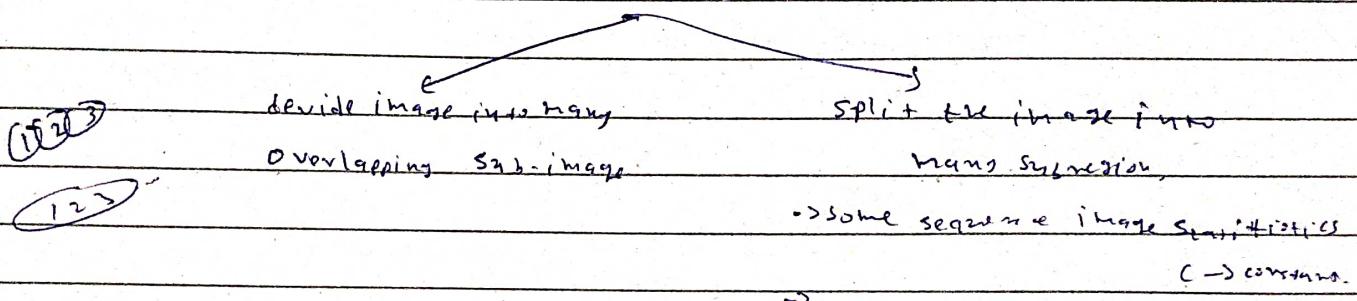
$$s_{\text{output}} = \begin{cases} s_1 & \text{if } 0 \leq f_i < t_1 \\ s_2 & \text{if } t_1 \leq f_i < t_2 \\ \vdots & \vdots \\ s_n & \text{if } t_n \leq f_i \leq b \end{cases}$$

* Adaptive thresholding

- It changes the thresholding dynamically over the image.
- It is also known dynamic threshold algo.
- Adaptive thresholding works well in situations where image is affected by non-uniform illumination.

Technique to apply

Threshold algo



* Optimal Otsu thresholding algo.

- It is used where there is a considerable overlapping of the histogram.
- It uses merit function of either maximization or minimization to determine the optimal threshold.