* Image compression

-> Image compression refers to the Process of redundancy amount of data required to represent the given quantity of information for digital image. The basis of reduction Process is removeal of redundant data.

methods
1. Huffman coding
2. Arithmatic & Transform coding

original image → Compression Data

$f(x,y)$ → Source Encoder → Channel Encoder → Channel → Channel Decoder → Source Decoder → $\hat{f}(x,y)$

Source Encoder: Removes redundancy
Channel Encoder: Increases noise immunity

Stages of Encoding :-

$f(x,y)$ → Mapper → Quantizer → Symbol Encoder → $\hat{f}(x,y)$

Mapper => Reduces interpixel redundancy, reversible operation
Quantizer → Reduces Physcovisual Redundancy, non - reversible op^n.
Symbol Encoder → Create a fixed or variable length code, Reversible

Source Decoder :-

Channel → Source Encoder → Inverse mapping → $\hat{f}(x,y)$

* Redundancy : Redundancy means Repetitive data

Redundancy
- Statistical
  - Inter-pixel
    - Spatial → correlation b/w neighbouring pixel values
    - Temporal → correlation b/w adjacent frames in a seq^n of image in vid
  - Coding
- Psychovisual

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

Coding Redundancy = Avg. bits used to code − Entropy

$$\boxed{CR = \sum_{k=0}^{n} l(r_k)\,P(r_k) - \sum P_i \log_2\left(\frac{1}{P_i}\right)}$$

length of code   Prob   (gray level)

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

→ It solved by algo. 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 decompresse 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~~ an image without any quality loss. eg. Huffman, Arithmatic
- remove unnecesary metadata from JPG & PNG.

(2) Lossy :

- Reconstructed data is an approximation to the original data

eg. linear Prediction & Transform coding diff

lossless compression	Lossy compression
-> file can be restored in its original	- Not restored
-> Quality doesn't compromised	- compromised
-> It does not reduce the size	-> reduce the size of data
-> ex. Huffman & Run length	-> ex Transform coding & wavelet
- It used in text, image, sound	-> It used in images, audio, video
-> does not eliminate	-> 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
  - Achive 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
a_3	0.1	101	0.1	0.2
a_4	0.1	1001	0.1	0.1
a_5	0.06	10000	0.1	
a_6	0.04	10001		

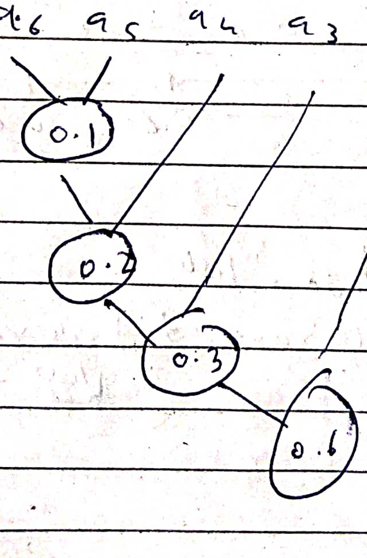
0.1, 0.1 → 0.2

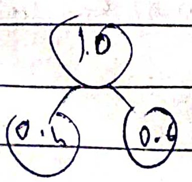
0.2, 0.1 → 0.3

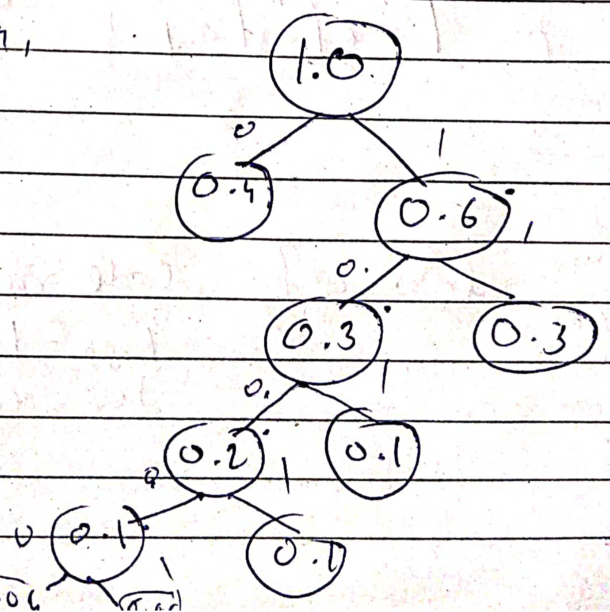
0.3, 0.3 → 0.6

0.6, 0.4 → 1

① ~~0.04~~ a_6 a_5 a_4 a_3 a_2 a_1







L = Avg. Length of code = $\sum P_k l_k$

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

= ans

$H(X)$ = Entropy = $\sum_{k=1}^{n} P_k \log_2 \frac{1}{P_k} = 0.4\times\log_2\frac{1}{0.4} + 0.3\times\log_2\frac{1}{0.3} + \dots + 0.04\times\log_2\frac{1}{0.04}$

$= 0.159 +$

= ans

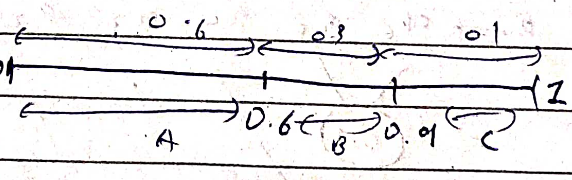
Coefficient $\eta = \frac{H}{L}\times 100$ → ans in %.

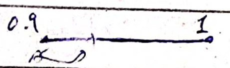
☆ Arithmatic coding: It is an alternative to huffman compression it enables characters to be represented as fractional bit lengths.

eg. code CAB using arithmatic coding.

Character	A	B	C
P	0.6	0.3	0.1

(1) Devide the ranges into 0-1





$A = 0.9 + 0.1\times0.6 =$

$B = 0.9 + 0.1\times0.3$

$C = 0.9 + 0.1\times0.1$

formula → Code range = high - low

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

low = low + code range × ~~high~~ low range (symbol)

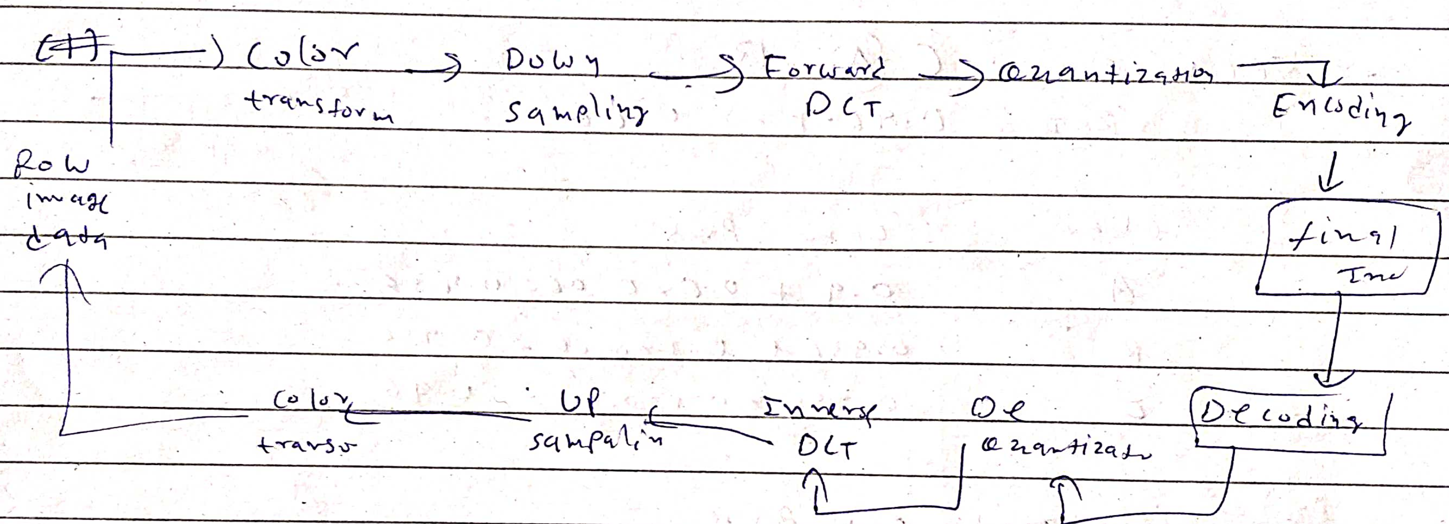
JPEG / / frequency based compression

→ 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 adjected.

→ It is followed by 8 steps.



Step 1: Input image splite 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 fn to convert into frequency domain

$$F(u,v) = \frac{C_u}{2}\frac{C_v}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} f(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$C_u = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u=0 \\ 1, & \text{if } u>1 \end{cases}$$ similarly for C_v

$$f(x,y) = \sum_{u=0}^{7}\sum_{v=0}^{7} F(u,v)\frac{C_u}{2}\frac{C_v}{2}\cos\left[\frac{(2x+1)u\pi}{16}\right]$$

split → why (PCE) → DCT → quantization → zigzag → vectoring → RLE → Huffman

Step 4: quantization is used to reduce the no. of bits per sample.

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

Step 5: The zigzag scan is used to scan the 8×8 matrix to 1×64 vector. it used to group low frequency coefficient to top level of vector & high freq to the bottom

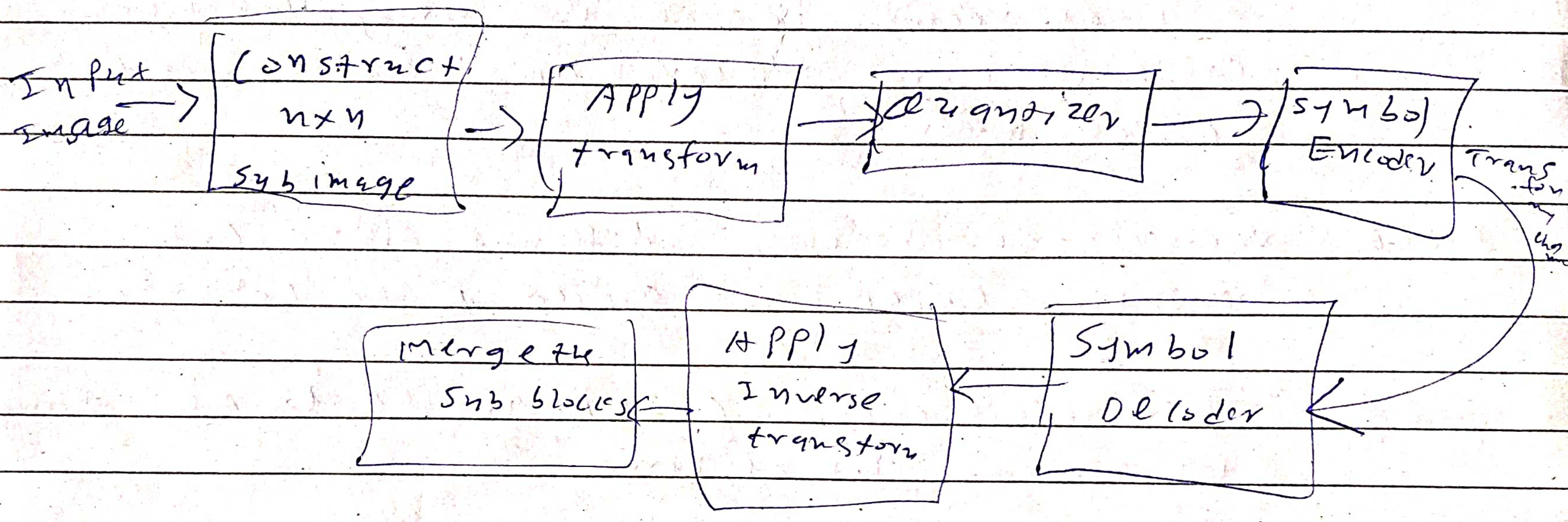
Step 6: vectoring is used for different pulse code modulation (DPCM) is applied to DC component

Step 7: run length coding (RLE) is applied to AC component for removing lot of zeros

Step 8: finally apply Huffman coding to the ~~image~~ vector and get ans

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 devided into a set of sub images.
- The values of n is a power of 2
- This step is necessary to reduce computational 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

→ Bit Allocation

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

→ Zonal coding

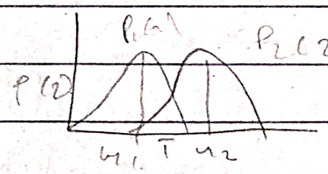
- multiplying each transform coefficient by the corresponding element in zonal mask

→ thershold 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 and background of image



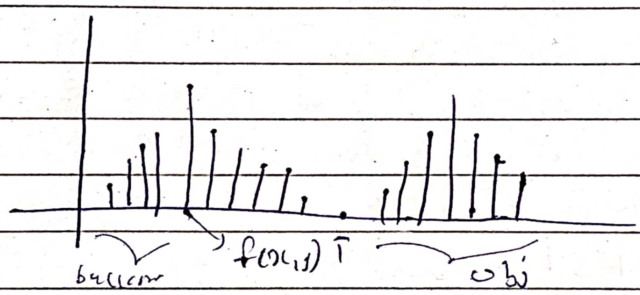
$P(z) = P_1 P_1(z) + P_2 P_2(z)$

$P(z) = \sum P_i P_i(z)$

Thresholding may be defined as
- single level thresholding.
- multi level thresholding.

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.

ex :



→ . 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

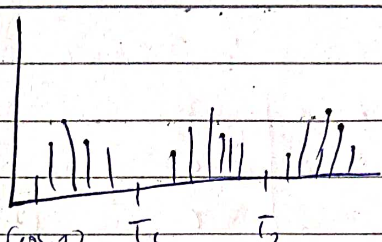
$f(x,y) > T$, It is called object

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

→ It called b/z we have separated info of an image by single thre

(2) Multilevel Thresholding.

-> More than one thresholding.



$f(x,y) > T_1 < T_2$ -> object

$f(x,y) < T_1$ -> background

$f(x,y) > T_2$ -> object

$E(T) = \int P_2(z)\, dz$

Error $= E(T) = P_2 E_1(T) + P_1 E_2(T)$

☆ Types of thresholding

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

only gray level

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

gray + 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

Problem minimize Image segment

☆ There is [If filter mask are given of any image then we can find the Threshold value by using that formula

$$\text{Threshold} = \frac{1}{2}(a_1 + a_2)$$

-> where a_1 and a_2 are the avg of pixel

-> We can devide in two group one is a_1 is greater then avg pixle value [take avg of grater than the avg. values]

-> a_2 = less than the avg. pixels value & take the avg.

-> then put in formula

Ex.

4	2	3
8	9	1
7	5	1

(i) $\frac{4+2+3+8+9+1+7+5+1}{9}$ = ans

a_1 (avg) $= \frac{\text{ans} \leq \text{pixels}}{\text{Total ans}}$ $a_2 = \frac{\text{ans} > \text{pixels}}{\text{Total ans}}$

Threshold = write formula