Unit 4

Lecture 05

Image Compression

- Coding redundancy, Interpixel redundancy, Psychovisual redundancy, Huffman coding, Arithmetic coding, lossy compression techniques, JPEG compression.
- The art and science of reducing the amount of data required to represent an image, is data compression. It is one of the most useful and commercially successful technologies in the field of DIP.
- Image compression is very important task in image processing.
- Images and Videos require lots of space for storage and take large transmission time.
- Data compression is the process of encoding data so that it takes less storage space or less transmission time than it would if it were not compressed.

Data Compression :-

It is the mathematical process of ~~transmission~~ Transforming data to a smaller representation from the original.

Original data
N1 Bits

Compressed Data
N2 Bits

① If N1 = N2 → no compression
② If N2 < N1 → significant compression
③ If N1 < N2 → Data explosion.

- Data compression refers to the process of reducing the amount of data required to represent a given quantity of information.

- data (raw) & information (meaningful) are not same.

-

Types of Data in Image Processing (Multimedia data)

① Text data (present in flat file & understood by human)
② Binary data (M/C can interpret)
③ Image data (pixel data, contain intensity & Color info)
④ Graphics Data (vector form)
⑤ Sound data (audio information)
⑥ Video data (video information)

Why Data compression required ?

1. Storage (resource)
2. Transmission (image size)
3. faster Computation

Application of Data Compression –

1. Personal communication like voicemail & telephony
2. Computer networks – Internet
3. Multimedia applications
4. Image & signal processing
5. Digital & satellite TV
6. Video conferencing and digital library

Compression Scheme

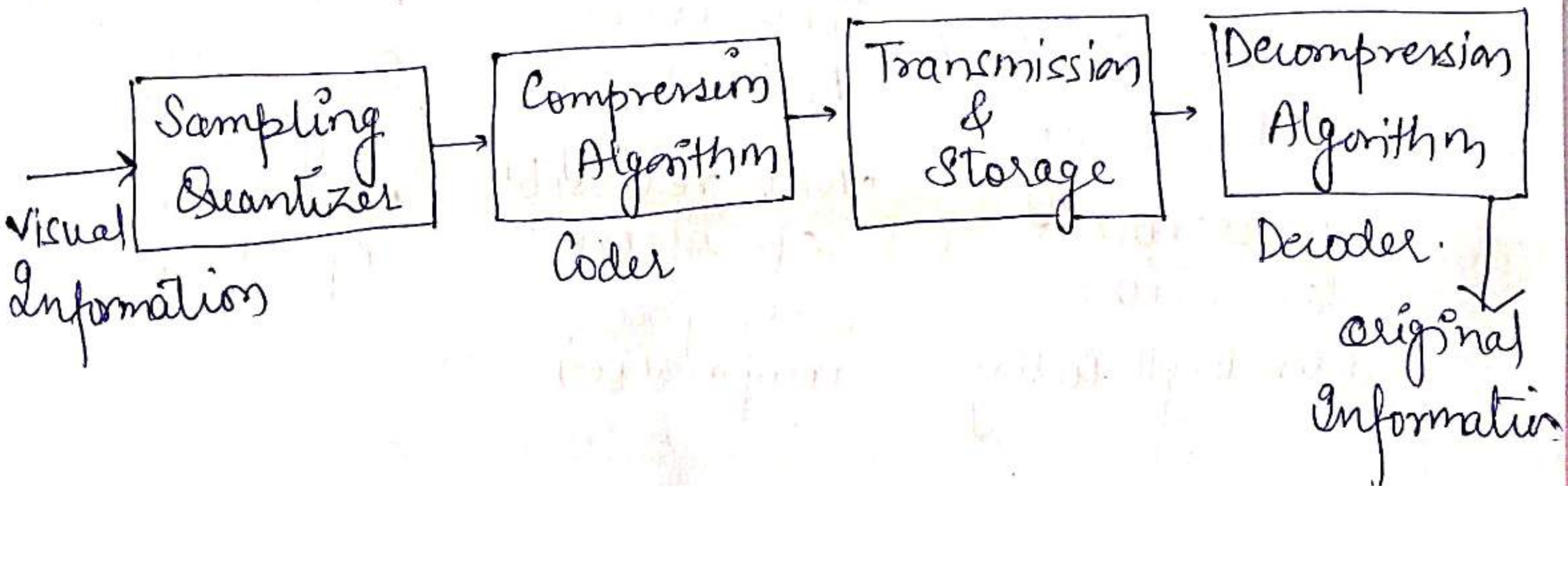


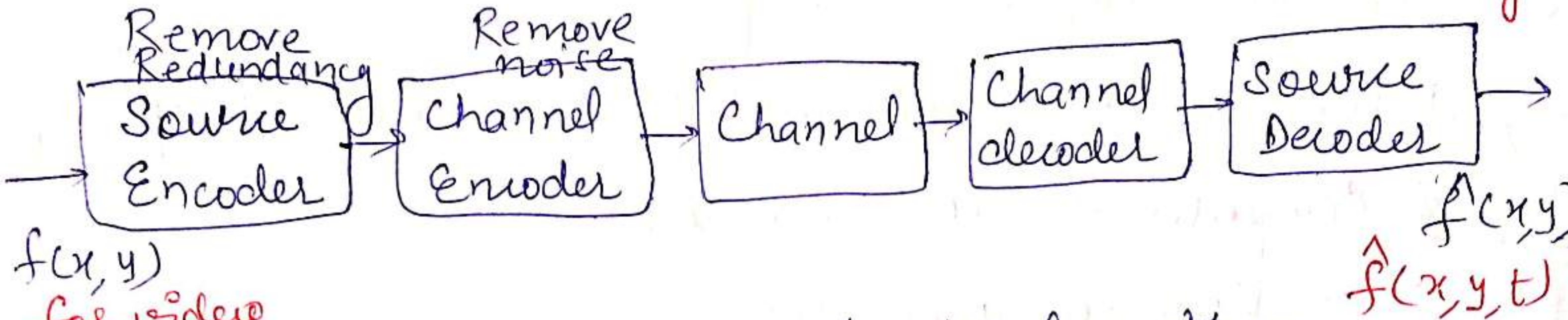
Image Compression Model

There are 2 main components of image compression model

(1) Encoder (compression)

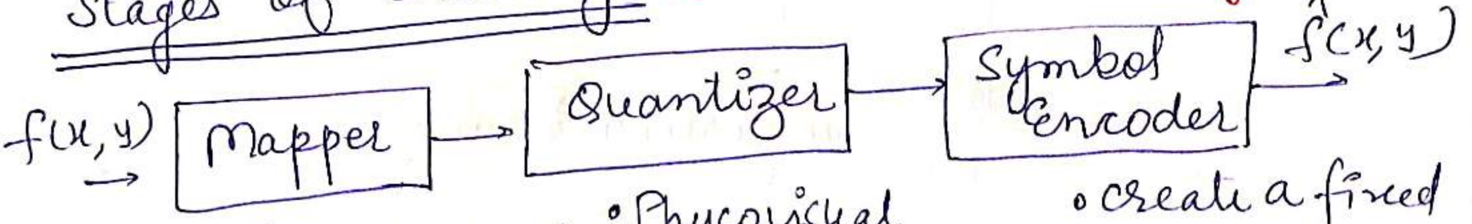
(2) Decoder (decompression)

Codec → device/program that perform both encoding & decoding.



f(x,y_t) - If there is no noise/noise free then channel encoder & channel decoder can be removed.

Stages of encoding (remove redundancy)



Mapper
- Reduces Interpixel Redundancy
- Reversible operation

eg. Run-length Coding

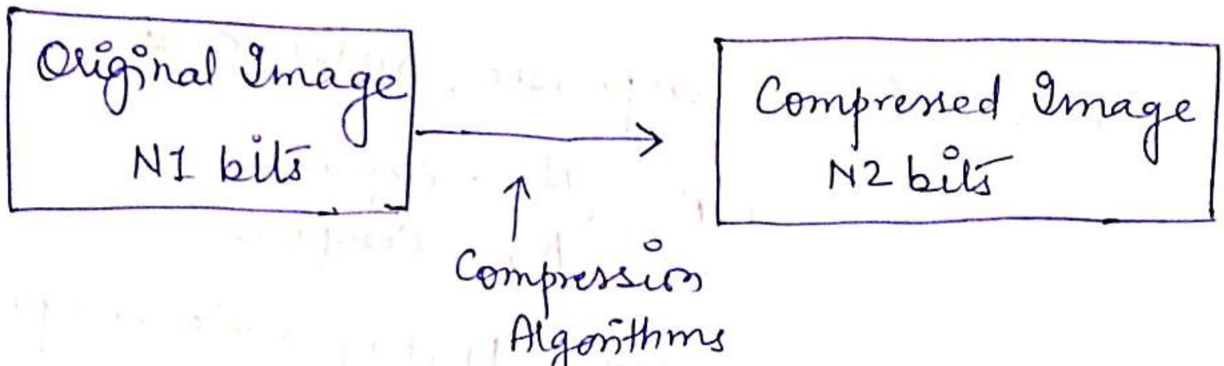
Quantizer
- Phycovisual Redundancy
- Not reversible operation
- error free compression

Symbol Encoder
- create a fixed or variable length code
- Reversible operation

Compression Measures :-

- Data Compression is the mathematical process of transforming data to a smaller representation from the original.

-



- Data Compression algorithms can be visualized as the mapping of as set of message symbols to codes using some logical rules of conversation.

eg: ① JPEG → Joint Photographic Experts Group
so its the logical compression → Talking abt data
② USA → United State of America

If Talking abt Image → that is called Physical Compression.

- Pixels or block of Pixels in image is → set of symbols
- Code → sequence of symbols or numbers.
- Code word → String of codes.

↓ visualized.

If we have N1 and N2 denote the number of bits in two representations of the same information and relative data redundancy (R_D) is represented as

$$R_D = 1 - \frac{1}{CR}$$

Where C.R is compression ratio and

$$CR = \frac{N_1}{N_2} \quad \begin{matrix} -\text{original} \\ -\text{compressed.} \end{matrix}$$

eg.. CR = 10:1 → means 1 bit of data compressed using 10 bits (original image) $R_D = 1 - \frac{1}{10} = .9$
↓
90% data is redundant.

Now check the CR.

(i) N1 = N2 $\quad CR = \frac{N_1}{N_2} = 1$

Means $\quad RD = 1 - \frac{1}{1} = 1-1 = 0$ so no redundancy
└→ i/p message is same as o/p message

(ii) $N_2 < N_1$ $\quad CR = \frac{N_1}{N_2} \approx \infty$ (very high)

Means $\quad RD = 1 - \frac{1}{CR} = 1$
└→ Compression is done.

(iii) $N_2 > N_1$

$CR = \frac{N_1}{N_2} = 0$ → Means Reverse compression
↓
So not interested yet

$R_D = 1 - \frac{1}{CR}$

$= 1 - \infty = \infty$

Eg:- Original image of size 256×256 and gray scale image. After compression its size is 6554 bytes (N_2), find compression ratio.

Ans:. Size of original image (N_1) $= \frac{256 \times 256 \times 8 (\text{bits})}{8}$ bytes

$= 65536$ bytes

$$CR = \frac{N_1}{N_2} = \frac{65536}{6554} = 10 \text{ or } 10:1$$

So $$R_D = 1 - \frac{1}{C_R} = \frac{10-1}{10} = .9 \Rightarrow 90\%$$

- In 2D compression, 2D intensity array suffer from different type of redundancy.

Q. What do u mean by redundancy??

- Redundancy means repetitive data
- This may be data that share some common characteristics or overlapped information.
- Redundancy may be implicitly and explicitly.
  - implicitly → not directly explain
  - explicitly → stated directly

Eg:. ① AAAABB is string
{A-4, B-2} for large string and large repetation

Ex $I = \begin{bmatrix} 10 & 10 & 20 \\ 20 & 20 & 20 \\ 20 & 20 & 10 \end{bmatrix}$

Repetation

$\begin{bmatrix} 10-3 \\ 20-6 \end{bmatrix}$ → explicit

Implicit

$I = \begin{bmatrix} 00 & 11 \\ 00 & 10 \end{bmatrix}$ write in form of LSB & MSB

$\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ & $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

There are four types of Redundancies

(1) Coding Redundancy

(2) Inter-Pixel Redundancy / Spatial

(3) Psycho-visual Redundancy / Irrelevant

(4) Chromatic Redundancy

(1) Coding Redundancy :-

What is code?

- Code is a system of symbols (letters, number, bit)
- Code word - each piece of info. assigned a sequence of code symbols.
- no. of symbols in each code is its length.

- It's related to probability distribution. If PDF is high that intensity is high in image, and If PDF is low its intensity value is low in image.

- Coding redundancy increased due to poor selection of coding technique.

- Coding technique assigns a unique code for all symbols of message.

- Wrong choice of coding technique also creates unnecessary additional bits. These extra bits are called redundancy.

- So in this self information is generated, which depend on probability.

$$I = \log_2\left(\frac{1}{P}\right)$$

$$= -\log_2 P \text{ bits}$$

So we can say $I \propto \left(\frac{1}{P}\right)$ — if Probibity 1, info 0; if Prob 0, info 1

Coding Redundancy = Avg. bits used to code − Entropy

let image with a size $M \times N$, r_k intensity variable [0, L-1] and probability of r_k

$$P_r(r_k) = \frac{n_k}{M \cdot N} \qquad k = 0, 1, \ldots L-1$$

if no. of bits used to represent each value of r_k is $l(r_k)$ (or length of code used)

- Avg. no. of bits required to represent each pixel.

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) P_r(r_k)$$

$$\text{Entropy } (H) = -\sum_{i=1}^{n} P_i \log_2 (P_i)$$

- Total no. of bits required to code an image of size $M \times N$ = $M \cdot N \cdot L_{avg}$.

eg:- $M \cdot N = 256 \times 256$

r_k	$P(r_k)$	Code 1	$l_1(r_k)$
$r_{87} = 87$	0.25	01010111	8 bits
$r_{128} = 128$	0.47	10000000	8 bits
$r_{186} = 186$	0.25	11000100	8 bits
$r_{255} = 255$	0.03	11111111	8 bits.
$r_k = 0$ (for $k \neq 87, 128, 186, 255$)	0	—	

for Code 1

$$L_{avg} = 8 \times .25 + 8 \times .47 + 8 \times .25 + 8 \times .03$$
$$= 8$$

(N1)

Original Image Size = $256 \times 256 \times 8$ = 524, 288 bit.

Let Code 2 for image

	Code 2	$l(r_k)$
r_{87}	01	2
r_{128}	1	1
r_{186}	000	3
r_{255}	001	3
r_k	-	-

$$L_{avg} = .25 \times 2 + .47 \times 1 + .25 \times 3 + .03 \times 3$$
$$= 1.81 \text{ bits}$$

Total no. of bits for entire image $= M.N.L_{avg}$
$= 256 \times 256 \times 1.81$
$= 118,621$ (N_2)

So, $CR = \frac{N_1}{N_2} = \frac{524,288}{118,621} \approx 4.42$

$R_D = 1 - \frac{1}{CR} = 0.774$ (means 77.4% data is redundant)

In this variable-length code is used r_{128} - most probable intensity (1 bit only)
r_{255} - least probable intensity (3 bits)

if fixed-length code (2 bit) = 4/1 10% less than variable length code.

(2) Inter-Pixel Redundancy :- / Spatial & Temporal Redundancy

- Inter pixel redundancy is related with the inter-pixel correlations within an image.
- Much of the visual contribution of a single pixel is redundant and can be gussed from the values of its neighbors.

Example:- Consider an image with a constant background.

- The visual nature of the image background is given by many pixels that are actually no required or necessary.
- This is known as Spatial redundancy or geometrical redundancy.
- Spatial redundancy may be present in
  - (i) Single frame (intra-frame)
  - (ii) Among multiple frames (inter-frame or temporal redundancy)

So, this type of redundancy (spatial) using down sampling.
or by using quantization, fix no. of bits are used to represent the pixel.

- Interpixel dependency is solved by algorithms like Predictive coding, bit plane algorithm, run-length coding and dictionary based algorithm.

(3) Psycho-Visual Redundancy :- / Irrelevant

In image processing some images are represented and don't show meaningful information. Human Eye can't extract useful information and this type of redundancy is called psycho-visual redundancy.

- The eye and brain do not respond to all visual information with same sensitivity.
- Some information is neglected during the processing by the brain. Elimination of this info. does not affect the interpretation of the image by the brain.

- Edges and textural regions are interpreted as important features and brain groups and correlates such groups to produce its perception of object.

- So psycho visual redundancy is distinctly vision related, and its elimination does result in loss of information.

- Quantization is an example of psyo visual redundancy.

- When 256 levels are reduced by grouping to 16 levels, objects are still recognizable. The compression is 2:1, but an objectionable graininess and contouring effect results.

- Example → Gray image is given (below eg.) represent by 8-bits value so size of image is

$256 \times 256 \times 8$ → it's gray ~~lev~~ image so we can represented by 8-bits only so

$$\text{Compression} = \frac{256 \times 256 \times 8}{8} = \frac{65,536}{1}$$

(4) Chromatic Redundancy :-

- It refers to the presence of unnecessary colors in an image.
- The color channels of color images are highly correlated and human visual system can not preceive millions of colors.
- Therefore the colors that are not perceived by human visual system can be removed without affecting the image quality.
- Removal of irrelevant information involves a loss of real image information.
- fidelity - Accurate reproduction of data
- Distortion - Reconstructed image from the original.

(1) Objective fidelity

Error, SNR, PSNR

if $f(x,y)$ - original

$\hat{f}(x,y)$ - compresed

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

- It's a simple and convenient way to evaluate information loss,

(2) Subjective fidelity

- This can be done by presenting a decompressed image to a cross section of viewers and averaging their evaluations

eg: Scaling

-3, -2, -1, 0, 1, 2, 3

worse, better, good, excellent

Compression Algorithms and its types

- The objective of Compression algorithm is to reduce the source data to a compressed form and decompress it to get the original data
- Compression Algorithm has two components

(1) Modeller
* It is used to compress the data using the knowledge of data
* Present both side (sender & receiver)
* It can static (models same at both side) or dynamic (models can be changed according to data).

(2) Coder
* Sender side coder is encoder and receiver side decoder
* Coder codes the symbols independently and decoder decodes the msg from the compressed data.

- If Modeller same at Sender and Receiver side than this is called Symmetric Scheme
- Otherwise it is Assymetric scheme

Compression Algorithm

Lossless
- Reconstructed data is identical to original data (Entropy coding)
eg:- Huffman, Arithmetic, Shanon-fano coding
- Mostly used in medical & legal domain

Lossy
- Reconstructed data is an approx. to original data (Source Coding)
eg: Linear Prediction, Transform Coding
- Used in broadcast, TV and MM.

Difference b/w lossless & lossy Compression

1. Reversible Process	1. Non reversible process
2. No info. lost	2. Some info. lost
3. Compression ratio less	3. Compression ratio high
4. Uses for data that human can handle directly like - Text data	4. Used for diffused data that human can't understand
5. Compression independent on psychovisual system	5. Compression is dependent on psychovisual system

Lossless Compression Algorithm

Huffman Coding :-

- It is a type of variable length coding
- In this coding redundancy can be eliminated by choosing a better way of assigning the codes.

Huffman coding Algorithm:

1) List symbols & sort probabilites per symbol
2) Combine lowest two probabilites of symbols and label new code with it.
3) Newly created item is given priority and placed at the highest position in the Sorted list.

4. Repeat step 2 until one node remain

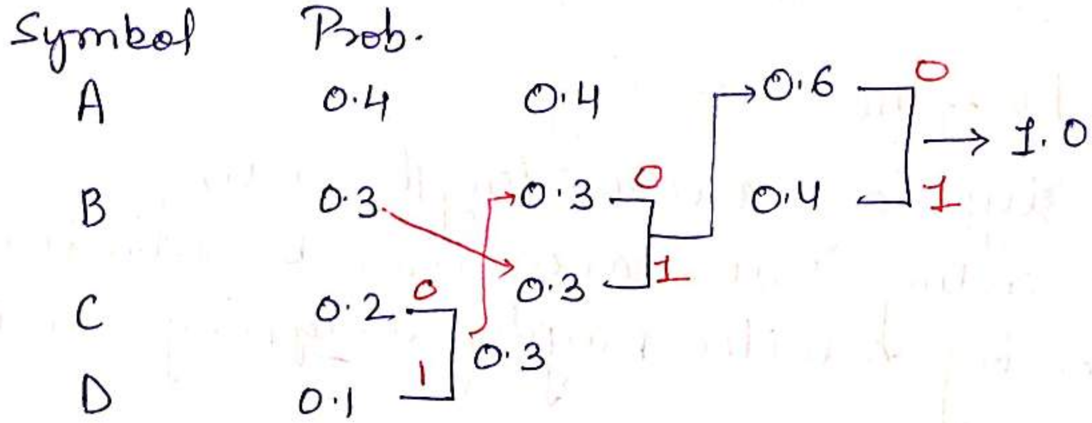
5. Assign code 0 to higher up symbol and 1 to the lower down symbol

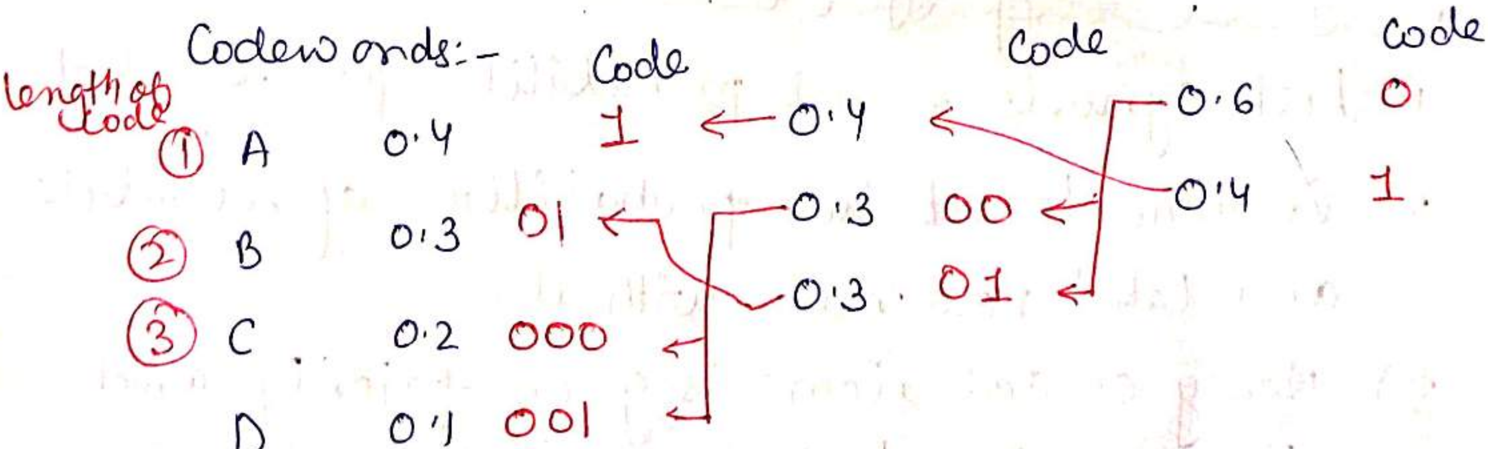
6. Now trace the code symbols going backwards.

Example: Calculate Huffman code for the set of symbols as shown in table.

Symbols.	A	B	C	D
Prob.	0.4	0.3	0.2	0.1

Solu:-





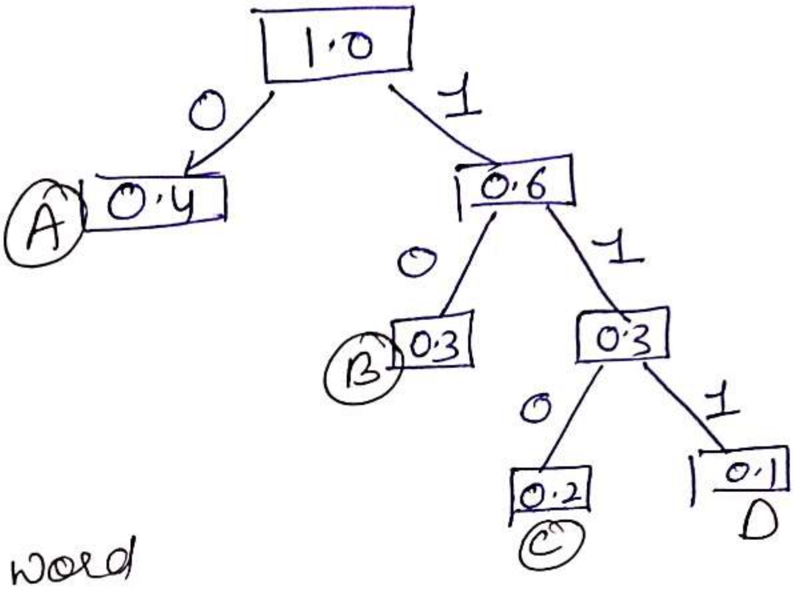
$L_{avg} = 0.4\times1 + 0.3\times2 + 0.2\times3 + 0.1\times3$

$= 0.4 + 0.6 + 0.6 + 0.3$

$= 1.9$ bits/pixel

Huffman Code Tree Approach

A	B	C	D
0.4	0.3	0.2	0.1



Code word

A - 0

B - 10

C → 110

D → 111

eg. find the Huffman coding for below table

a_2	a_6	a_1	a_4	a_3	a_5
0.4	0.3	0.1	0.1	0.06	0.04

Arithmetic Coding :- Lossless Compression

- lossless compression is prefrable becausze data is not lost (used for preservation of data)
- CR is smaller than lossy compression.
- In AC technique, a string of characters like the word "hello there" is represented using a fixed no. of bits per character, as in ASCII code.

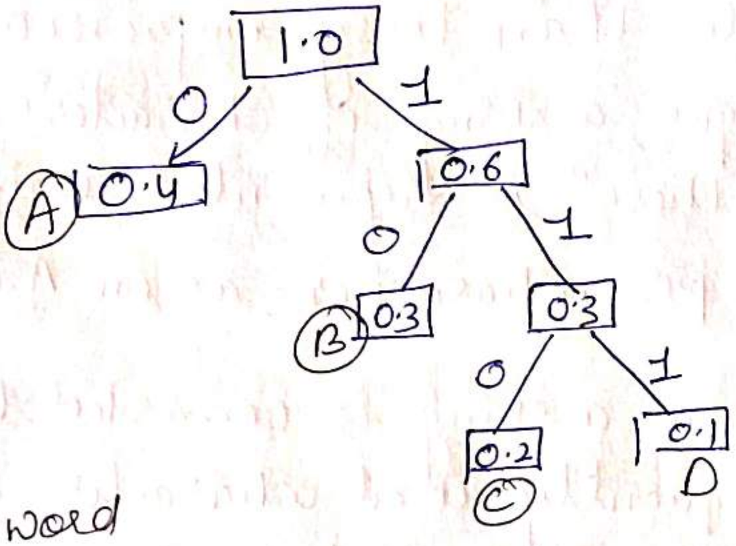
- Example when a string is converted to arithmetic endocing, frequently used characters are stored with fewer bits and not-so-frequently occurring characters are stored with more bits, resulting in relatively fewer bits used in total.

How Arithmetic coding different from Huffman coding

Arithmetic Coding	Huffman Coding
1. Complex technique for Coding short message.	1. Simple Technique
2. Gives optimum result	2. Does not give optimum result. (variable length)
3. There is no one to one correspondence b/w source symbol and code word	(a, b, c → find value) 3. There is one to one Correspondence b/w source Symbol and code word
4. Compression ratio is more	4. Compression ratio is less
5. Execution time is more	5. Execution time is less

Huffman Code Tree Approach

A	B	C	D
0.4	0.3	0.2	0.1



Code word

A - 0

B - 10

C → 110

D → 111

Eg. find the Huffman coding for below table

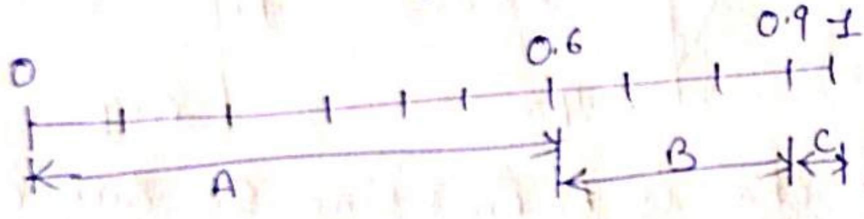
a_2	a_6	a_1	a_4	a_3	a_5
0.4	0.3	0.1	0.1	0.06	0.04

Q:- Code the string CAB using arithmetic coding

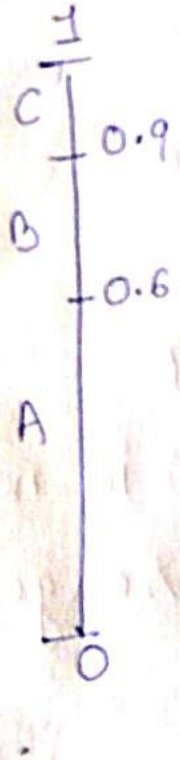
Char	A	B	C
Prob	0.6	0.3	0.1

<u>Ans</u>:-

<u>Step1</u>:- Divide the range into 0 to 1 (based on probability)



- In string first character is C and its range is <u>0.9 to 1</u>
- Code would start in this range only
- Divided among symbols according to prob.



first char is C in string so take range of C (0.9 − 1)

diff = 0.1

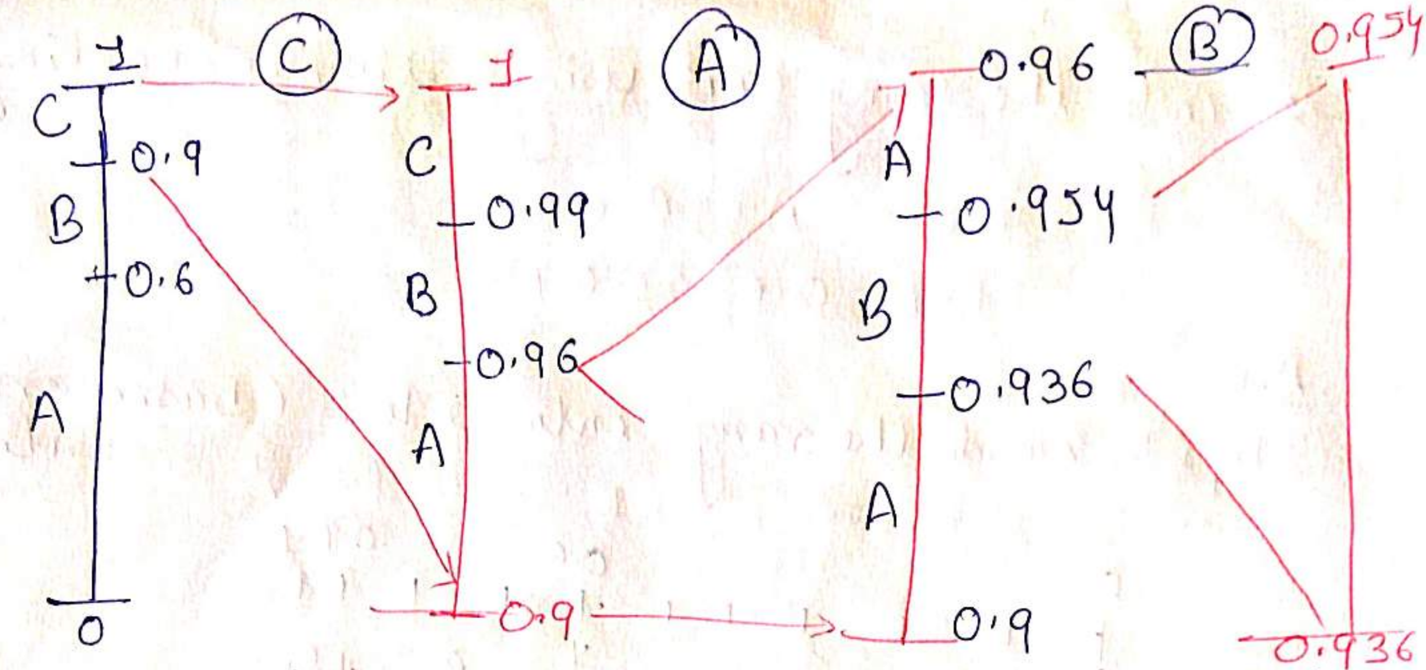
Now calculate A, B, C probability

A = 0.1×0.6 = 0.06 + 0.9 (range of C)
= 0.96

B = 0.1×0.3 = 0.03 + 0.96
= 0.99

C = 0.1 × 0.1 = 0.01 + 0.99
= 1

Now draw it



Now next char is A in string so range of A is
0.9 to 0.96

Diff b/w 0.9 − 0.96 = 0.06

So calculate probability of A, B, C in this new range (range of A)

A = 0.06 × 0.6 = .036 + 0.9
= 0.936

B = 0.06 × 0.3 = .018 + 0.936
= 0.954

C = 0.06 × 0.1 = 0.006 + 0.954
= 0.96 Draw in new scale

Now next char is (B) and its range is
(0.954 − 0.936)

⇒ Now add it and divide by 2

$$\frac{0.954+0.936}{2} = \frac{1.890}{2} = \boxed{0.945}$$

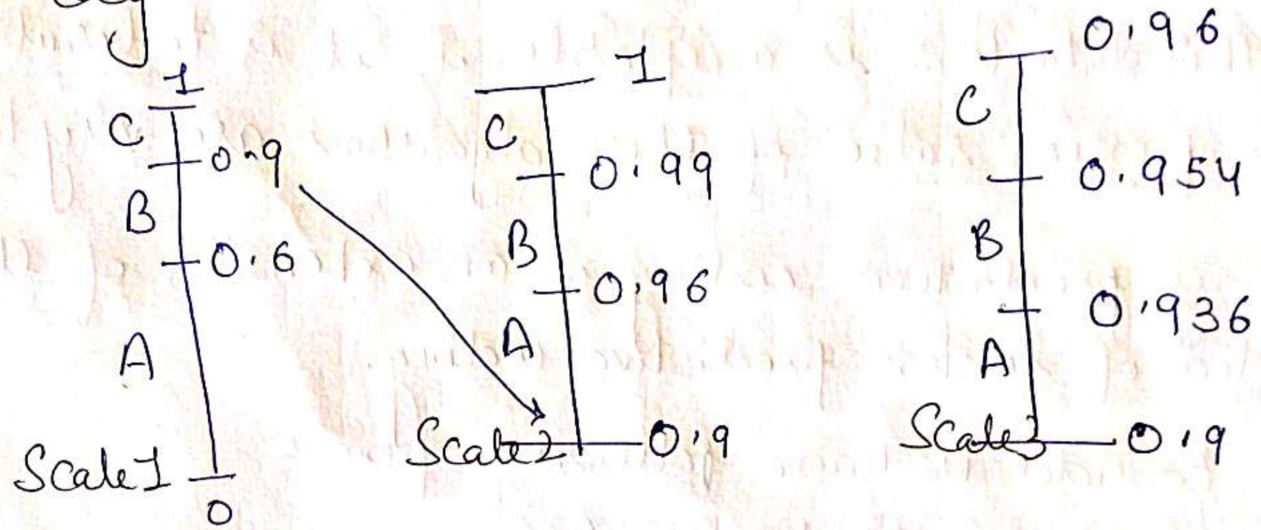
Tag

final code

Spatial and Temporal Redundancy: –

Tag = 0.945

Now find the sequence with the help of tag



Now find the sequence

0.945 in first scale

→ 0.945 in the range of C (0.9 – 1) so → C

→ 0.945 in scale2 (in A range 0.9 – 0.96) → A

→ 0.945 in scale3 → B (0.936 – 0.954) → B

So, the final sequence is

CAB Ans

Lossy Compression Techniques

- Lossy compression Algorithms incur a loss of information
- This loss is known as distortion
- This data loss is acceptable if it is tolerable
- Compression ratio of these algorithms are very large
- Lossy predictive coding is an extension of the idea of lossless predictive coding.

Eg:- Consider we have following pixels
{23, 64, 39, 47, 55, 63}

Value	Predictive coding	
		overloading
23	23	max 5 bits are required
64	(64-23) = 41 > 32	for coding $2^5 = 32$
39	39-64 = -25	6 bits required + 1 sign bit
47	47-39 = 8	- for all using 5 bit + 1 sign bit = 6
55	55-47 = 8	
63	63-55 = 8	- due to this loss of info. is there

- So, unlike the error-free approaches, lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression.

- Many lossy encoding techniques are capable of reproducing compressed by more than 100:1 whereas in loss-less compression it is seldom achieved more the 3:1
- The principle diff b/w loss-less and lossy approaches is the presence or absence of the quantizer block. (It takes advantage of interpixel redundancy)
- Quantizer is used in lossy compression only.
- With the help of delta modulation, we can extended the process by using only one bit to encode the difference b/w the adjacent pixels.
- So Delta modulation may be +ve or -ve

eq. for DM

$$\hat{f}(n) = \alpha \dot{f}(n-1)$$

where $\alpha \rightarrow$ prediction coefficient (normally 1)

eg:- for eg. for first digit

so, $\hat{f}(n) = \hat{f}(n-1)$

- Error is computed as

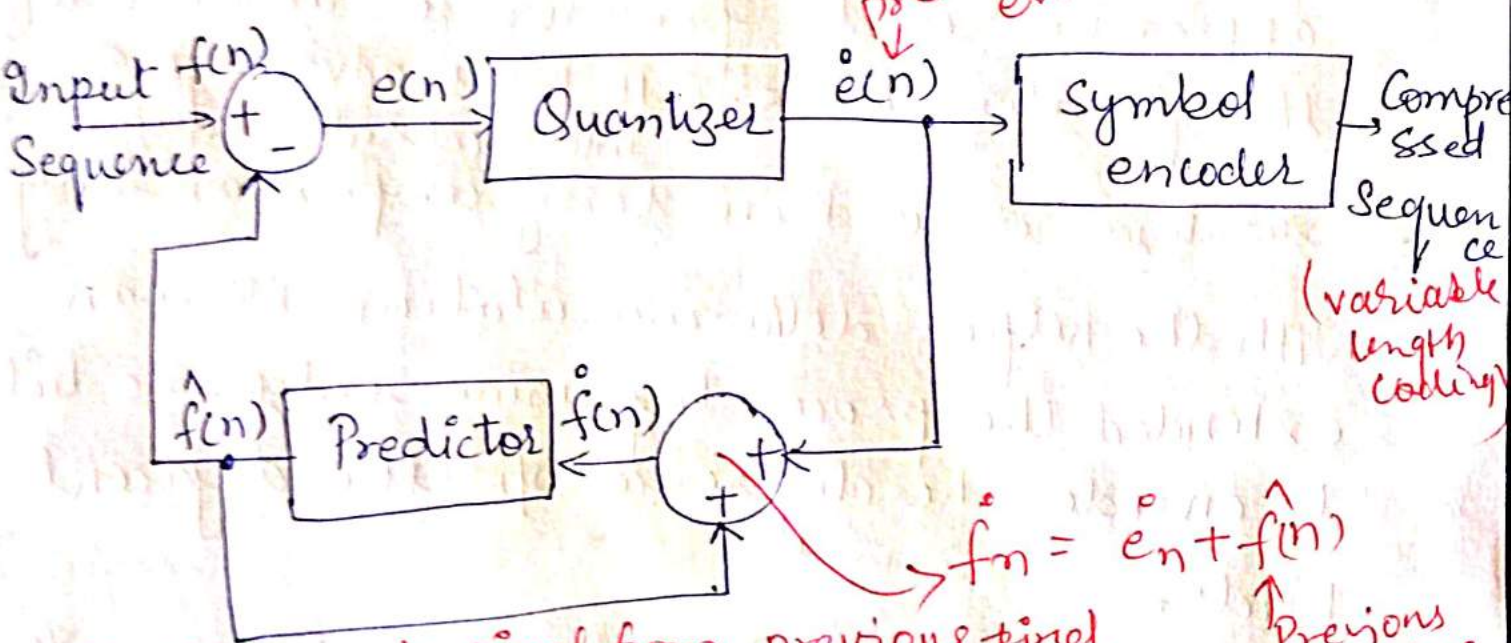
$$e_n = f_n - \hat{f}_n = f_n - \hat{f}(n-1)$$

- Error is quantized as

$$e_n = \begin{cases} +\varepsilon & \text{for } e(n) > 0 \\ -\varepsilon & \text{for other values.} \end{cases}$$

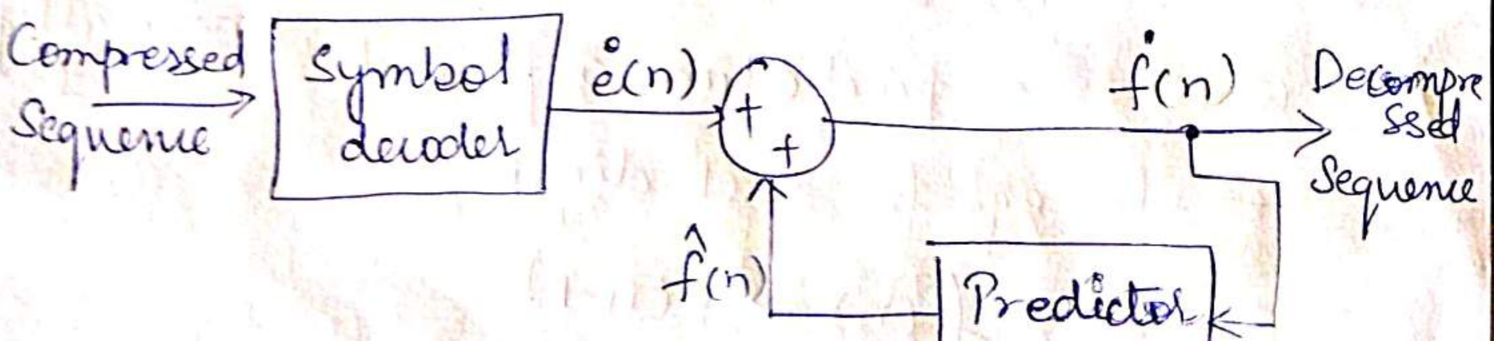
lossless predictive coding model

① encoder



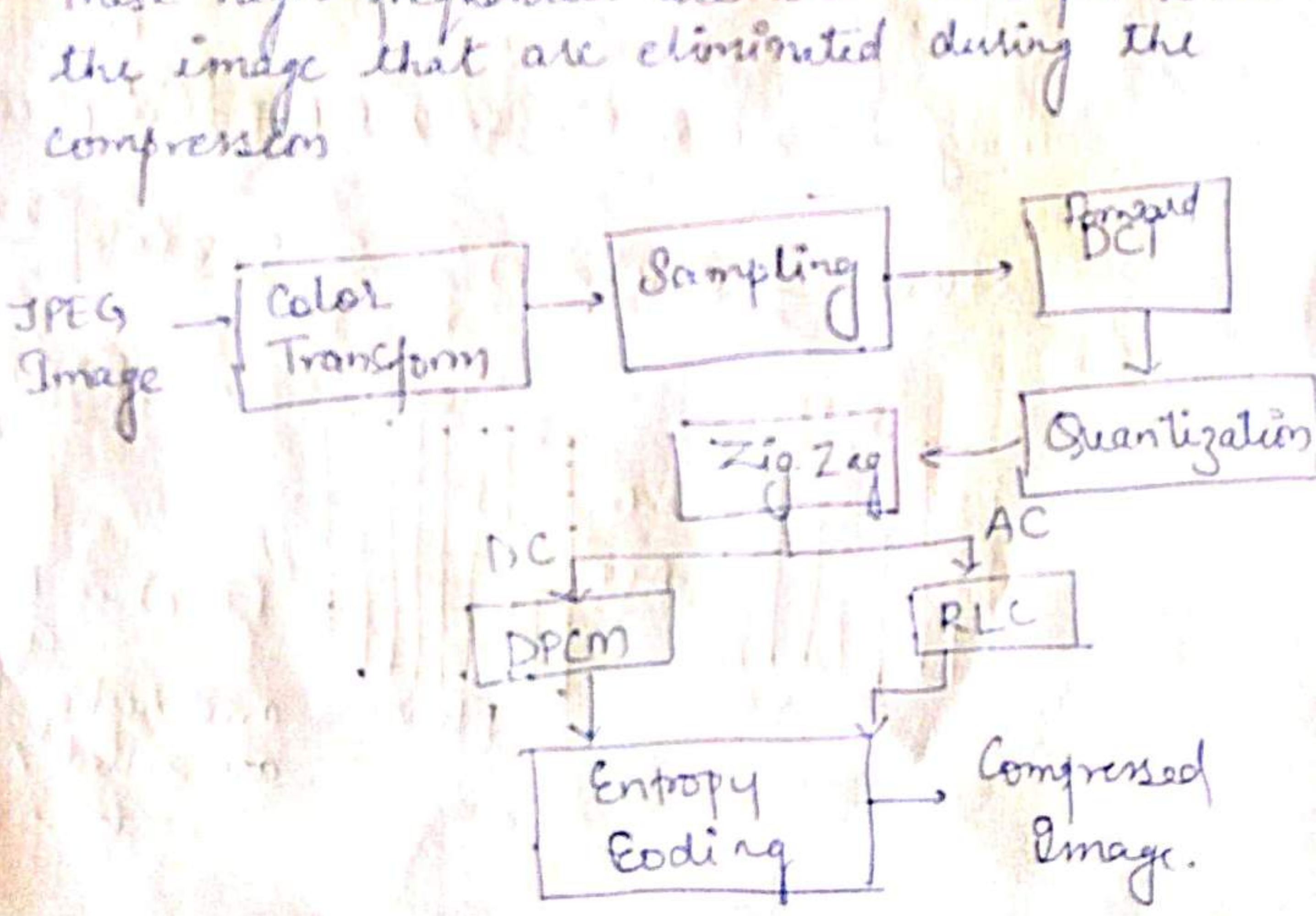
• Predict next pixel from previous pixel, encode only the diff from the actual and the predicted

② Decoder



JPEG Compression:-

- JPEG stands for Joint Photographic Experts Groups, it is a lossy compression algorithm for images
- A lossy compression scheme is a way to inexact represent the data in the image, such that less memory is used yet the data appears to be very similar.
- The JPEG algorithm takes advantage of the fact that humans can't see colours at high freq. These high frequencies are the data points in the image that are eliminated during the compression



Steps :-

1. The input image is divided into a small block which have 8x8 dimensions. This dimension is sum up to 64 units. Each unit of the image is called pixel.

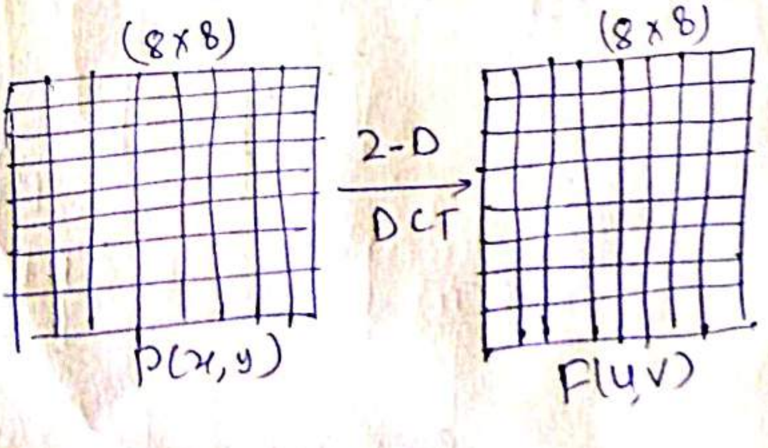
2. JPEG uses YC_bC_r model instead of RGB. So RGB image is converted into YC_bC_r model. (only Y component is used — Sampling)

3. It is forwarded to DCT (Discrete Cosine Transform). DCT uses cosine funⁿ and convert block of pixels from spatial domain to freq. domain (8x8 block)

$$f(w) = \frac{a(u)}{2} \sum_{n=0}^{N-1} f(n) \cos \frac{(2n+1) w \pi}{16}$$

image is 2D matrix, so 2-D DCT can be applied

$$G(u,v) = \frac{1}{4} \alpha(u) . \alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos \left[\frac{(2x+1) u \pi}{16}\right] . \cos \left[\frac{(2y+1) v \pi}{16}\right]$$



F(0,0) is DC component and rest of P(x,y) are called AC component

(4). Quantization

- By multiplying DCT matrix by some mask we can zero out elements of the matrix. (preserve low freq)
- Quantization is applied to reduce the no. of bits/sample.

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

(5) Zig-Zag-Scan

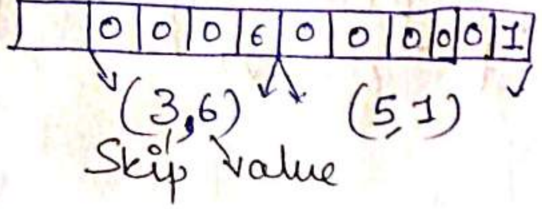
- Maps 8X8 matrix to a 1X64 vector
- Zig-zag make group of low feq coefficients at the Top of vector and high feq coffients at the bottom
- To remove large no. of zero in the quantized matrix, zig-zag matrix is used.

(6) Vectoring; DPCM on DC component

Differential Pulse Code Modulation applied on DC component. DPCM encodes the difference b/w the current block and previous block

(7) RLE on AC components

- Run length Encoding is applied on AC components
- 1X63 vector has lots of Zero.
- Encodes as (Skip, value) Skip → no. of zeros
  value, - non-zero value



(8) DC component coded into Huffman

DC & AC components finally need to be represented by a smaller number of bits.

- Huffman coding provide significant level of compression by replacing long strings of binary digits by a string of much shorter code words.
- (Table is already given)

DC
(Size, Amplitude)
Size ↓ no. of bits needed
Amplitude ↓ actual bits

AC
(Runlength/Size, Value)
Runlength → length of zero
Size → no. of bits needed
Value → Actual value of AC comp

- So, JPEG compression works with colored & gray image but not with binary images.
- Suitable for applications like, satellite, medical, general photography etc.