UNIT - 6
IMAGE COMPRESSION

Image compression addresses the problem of reducing the amount of data required to represent a digital Image.
→ The basis of the reduction process is the removal of redundant Data.

① Fundamentals :-

The term data compression refers to the process of reducing the amount of data required to represent a given quantity of information

→ The words data and information are not synonymous.
→ data are the means by which information is conveyed

ex:- A story is the information of interest it can be told by different number of words by two individuals. in that some may be essential and some may be non essential. and some times contain data which is already known.

This already known data is called data redundancy.

→ Data redundancy is a central issue in digital image compression.

If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information the relative data redundancy R_D of the first data set is defined as

$$\boxed{R_D = 1 - \frac{1}{C_R}}$$

where C_R is called compression ratio

$$\boxed{C_R = \frac{n_1}{n_2}}$$

If $n_2 = n_1 \Rightarrow C_R = 1 \quad R_D = 0$

when $n_2 << n_1 \Rightarrow C_R \to \infty$ $R_D \to 1$ (indicate significant compression)

when $n_2 >> n_1 \Rightarrow C_R \to 0$ and $R_D \to -\infty$ (indicate second data contain much more data than first.

In general C_R lie in open interval $(0, \infty)$

R_D lie in open interval $(-\infty, 1)$.

practical compression ratio such as 10 (10:1) means First data set has 10 information for every 1 unit of second data set.

-> Redundancy of 0.9 implies 90% of the data in the first data set is redundant.

(2) In digital Image compression three Basic data redundancies can be identified

1. Coding redundancy
2. Interpixel Redundancy
3. psychovisual redundancy.

(1) Coding Redundancy:-

Let us assume a discrete random variable r_k in interval $[0,1]$ represents the gray levels of an image and each r_k occur with probability of $P_r(r_k)$

$$P_r(r_k) = \frac{n_k}{n} \qquad k = 0, 1, 2, \ldots L-1$$

where L is the number of gray levels

n_k is number of times the k^{th} gray level appears in image

n is total number of pixels in the image.

If the number of bits used to represent each value of r_k is $l(r_k)$

The average number of bits required to represent each pixel is

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

Average length of the code words (L_{avg}) is found by summing the product of number of bits $[l(r_k)]$ and probability of occurrence of gray level $P_r(r_k)$

→ Total number of bits required to code an $M \times N$ image is $MN L_{avg}$.

Example:-

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	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

L_{avg} for code 1 is 3

L_{avg} for code 2 $= \sum_{k=0}^{7} l_2(r_k)\, P_r(r_k)$

$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02) = 2.7$ bits.

Compression Ratio $= \frac{n_1}{n_2} = \frac{3}{2.7} = 1.11$

Redundancy (R_D) $= 1 - \frac{1}{CR} = 1 - \frac{1}{1.11} = 0.099$.

In this example code 2 is one of choice, because it assignes fewer bits to the more probable gray levels than to the less probable ares and achieves data compression.

This process is called variabl-length coding.

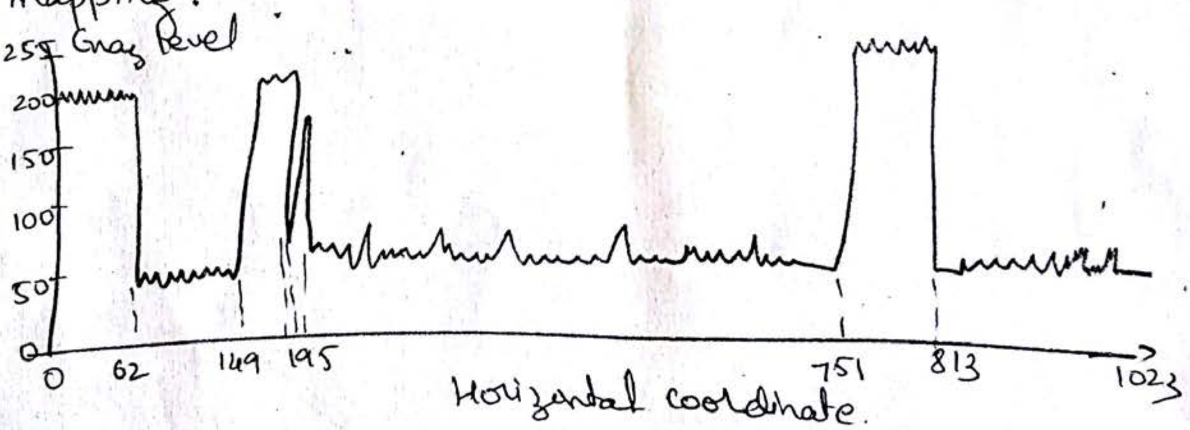
Interpixel Redundancy:-

Interpixel redundancy is the interpixel correlation within an Image.

→ The value of any given pixel can be predicted from the value of its neighbors.

→ Spatial redundancy, geometric redundancy and interframe redundancy are together refered as interpixel Redundancy.

→ Transformation of an image into 2-D pixel array normally used for human viewing and interpretation is called mapping.



line 100 :- (1, 63)(0, 87) (1,37) (0,5) (1,4) (0, 556)(1,62) (0, 210)

(Run-length code)

Psychovisual Redundancy:-

certain information simply has less relative importance than other information in normal visual processing and this information is called psychovisually redundant.

→ Psychovisual redundancy is different from other redundancies we discussed. it is associated with real or quantifiable visual information.

→ Its elimination is possible only because the information itself is not essential for normal visual processing.

→ the elimination of psychovisual redundant data in a loss of quantitative information so it is commonly referred to as quantization:

(3) Fidelity Criteria:-

Removal of psychovisual redundant data results in a Loss of real or quantitative visual information.

→ For assessment two general classes of criteria are used

(1) Objective fidelity criteria

(2) Subjective fidelity criteria.

Objective fidelity Criteria:-

When the level of information Loss can be expressed as a function of the original or input image and ~~the~~ the compressed and subsequently decompressed output image this is said to be based on an objective fidelity criterion.

Ex:- Root-mean-Square (rms) error between an input and output.

Let $f(x,y)$ represent an input image and $\hat{f}(x,y)$ denote estimate of $f(x,y)$ that results from compressing and Subsequently decompressing the input

The error $e(x,y)$ between $\hat{f}(x,y)$ and $\hat{f}(x,y)$ is defined as

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

Total error between two images is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]$$

where the images are of size $M \times N$

→ The root-mean-square (e_{rms}) between $f(x,y)$ and $\hat{f}(x,y)$ is defined as

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \right]^{1/2}$$

→ The mean-square signal-to-noise ratio of the output image denoted SNR_{ms} is

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2}$$

The rms value of the signal-to-noise ratio denoted as $SNR_{rms} = (SNR_{ms})^{1/2}$

<u>Subjective fidelity Criteria</u>:-

The evaluations made by using an absolute rating scale or by means of side-by-side comparisions of $f(x,y)$ and $\hat{f}(x,y)$.

The absolute rating scale such as $\{-3, -2, -1, 0, 1, 2, 3)$ to represent the ~~se~~ subjective evaluation {much worse

worse, slightly worse, the same, slightly better, better, much better} respectively.

Table:- Rating Scale of the Television Allocations Study Organization

Value	Rating	Description
1	Excellent	An image of extremely high quality as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable
4	Marginal	An image of poor quality: you wish you could improve it. Interference is some what objectionable.
5	Inferior	A very poor image, but you could watch it objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

4 Image Compression Models:-

A compression system consists of two distinct structural blocks an Encoder and a decoder.

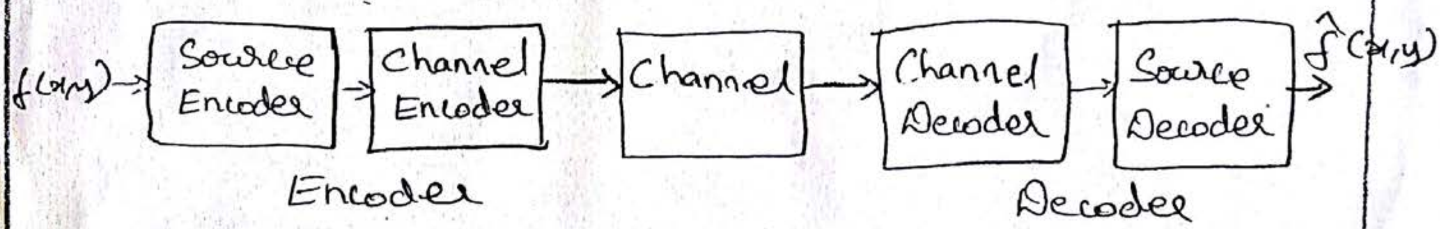


Fig:- A General compression System Model.

→ An input image f(x,y) is fed into the encoder, which creates a set of symbols from the input data.

→ After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image $f(x,y)$ is generated.

→ Both the encoder and decoder consist of two relatively independent functions or subblocks.

→ The encoder is made up of a source encoder which removes input redundancies and a channel encoder which increases the noise immunity of the source encoders output

→ The Decoder includes a channel decoder followed by a source decoder.

→ If the channel between the encoder and decoder is noise free then encoder and decoder are omitted.

(5) The Source Encoder and Decoder:-

The source encoder is responsible for reducing or eliminating any coding, interpixel or psychovisual redundancies in the input image.

→ The source encoder can be modeled by series of three independent operations

Source Encoder Model.
Fig:- Source Encoder

Each operation is designed to reduce one of the three redundancies.

In the first stage of the source encoding process, the mapper transforms the input data into a format designed to reduce interpixel redundancies in the input image

→ The operation is reversible. an example of a mapping is run-length coding.

→ The second stage or Quantizer block which reduces the accuracy of the mapper's output.

→ This Stage reduces the psychovisual redundancies of the input image.

→ This operation is irreversible and this must be omitted when error free compression is desired.

→ The third and final stage of the source encoding process the Symbol coder creates a fixed - or variable - length code to represent the quantizer output and maps the output in accordance with the code.

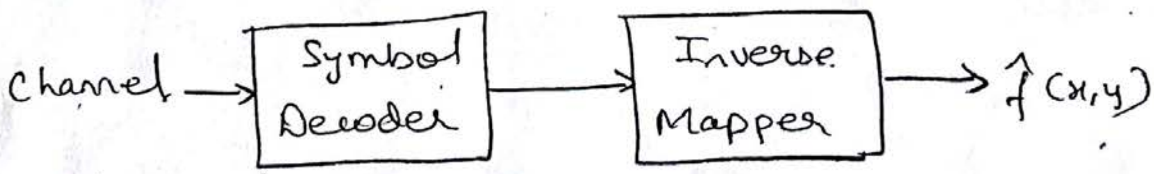
This stage reduces the coding redundancies and it is a reversible operation.

Note:- all the three stages not necessarily include in every compression system.

→ In predictive compression systems the mapper and quantizer are often represented by a single block.

The Source Decoder:-

The source decoder contains only two components a symbol decoder and an inverse mapper.



Source decoder

Fig:- Source decoder model

These blocks perform in reverse order the inverse operations of the source encoders i.e. symbol ~~endo~~ encoder and Mapper blocks.

→ Because quantization results in irreversible so it is not included in general source decoder Model.

The Channel Encoder and Decoder:-

The channel encoder and decoder play an important role in the overall encoding - decoding process when the channel is noisy.

→ They are designed to reduce the impact of channel noise by inserting a controlled form of redundancy into the source encoded data.

→ One of the most useful channel encoding techniques is Hamming code. devised by R.W. Hamming (1550).

Ex:- 3 bits of redundancy are added to a 4-bit word so that the distance between any two valid code words is 3, all single-bit errors can be detected and corrected

→ The 7-bit Hamming (7,4) code word $h_1 h_2 \ldots h_5 h_6 h_7$ associated with a 4-bit binary number $b_3 b_2 b_1 b_0$ is

$$h_1 = b_3 \oplus b_2 \oplus b_0 \qquad h_3 = b_3$$

$$h_2 = b_3 \oplus b_1 \oplus b_0 \qquad h_5 = b_2$$

$$h_4 = b_2 \oplus b_1 \oplus b_0 \qquad h_6 = b_1$$

$$h_7 = b_0$$

($h_1 h_2 h_4$ are even parity bits)

where $\oplus$ denotes exor operation

To decode a Hamming encoded result the channel decoder must check the encoded value for odd parity over the bit fields in which even parity was previously established.

$$c_1 = h_1 \oplus h_3 \oplus h_5 \oplus h_7$$

$$c_2 = h_2 \oplus h_3 \oplus h_6 \oplus h_7$$

$$c_4 = h_4 \oplus h_5 \oplus h_6 \oplus h_7$$

If a non zero value is found the decoder simply indicate the ~~po bit~~ error bit position and it is complemented. The

6 Error-Free Compression:-

Error-Free compression is used in application like medical and business documents where Lossy compression is prohibited.

→ Error-free compression technique generally are composed of two relatively independent operations

1. devising an alternative representation of the image in which its interpixel redundancies are reduced (Mapper)
2. coding the representation to eliminate coding redundancies. (Symbol encoder).

Variable-length coding:-

The simplest approach to error-free image compression is to reduce only coding redundancy.

→ Variable-length code assigns the shortest possible code words to the most probable gray levels.

Huffman coding:-

The most popular technique for removing coding redundancy is due to Huffman.

→ Huffman coding yields the smallest possible number of code symbols per source symbol.

→ The first step in Huffman's approach is to create a series of source reductions by ordering the probabilities of the symbols and combine the ~~lowest~~ probability symbols into a single symbol that replaces them in the next source reduction.

→ At the far left, a hypothetical set of source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values.

→ The second step in Huffman's procedure is to code each

reduced source starting with the smallest source and working back to the original source.
The minimal length binary code for a two-symbol source, is 0 and 1.

original source			Source reduction			
Symbol	probability	code	1	2	3	4
a_2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0
a_6	0.3	00	0.3 00	0.3 00	0.3 00	0.4 1
a_1	0.1	011	0.1 011	0.2 010	0.3 01	
a_4	0.1	0100	0.1 0100	0.1 011		
a_3	0.06	01010	0.1 0101			
a_5	0.04	01011				

The average length of this code is.

$$L_{avg} = (0.4)1 + (0.3)2 + (0.1)3 + (0.1)4 + (0.06)5 + (0.04)5$$
$$= 2.2 \text{ bits/symbol.}$$

Huffman's procedure are.

(1) probability subject to the constraint that the symbols be coded one at a time

(2) it is called a block code because each source symbol is mapped into a fixed sequence of code symbols

(3) It is instantaneous because each code word in a string of code symbols can be decoded without referencing succeeding symbols.

(4) it is uniquely decodeable, because any string of code symbols can be decoded in only one way.

Arithmetic Coding :-

→ Arithmetic Coding generates nonblock codes

→ An entire sequence of source symbols is assigned a single arithmetic code word.

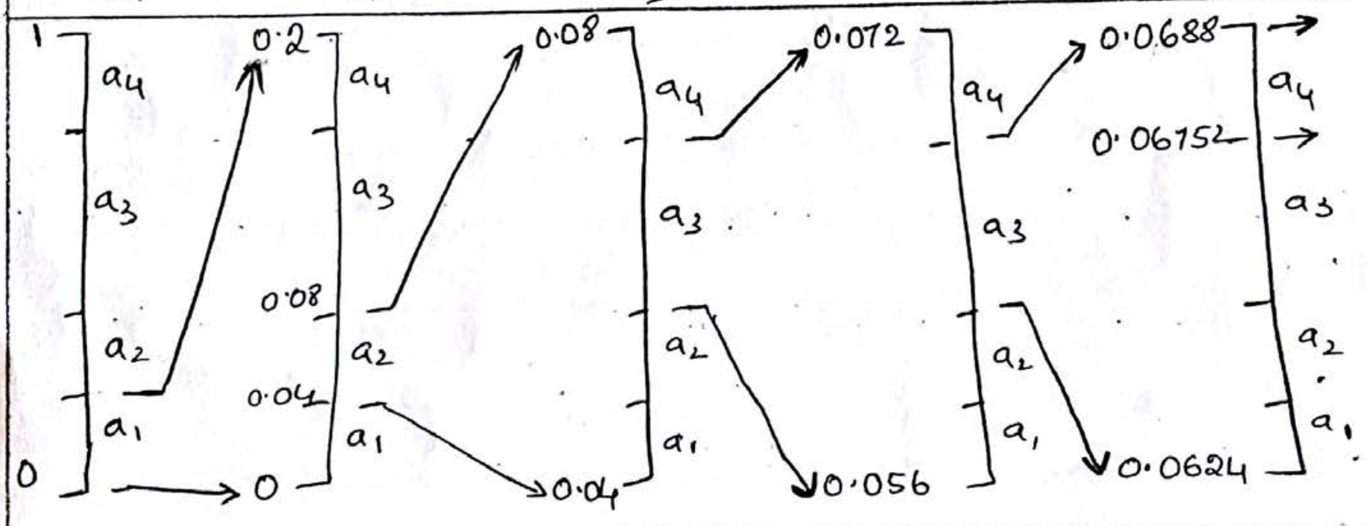
→ The code word itself defines an interval of real numbers between 0 and 1.

→ As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units required to represent the interval becomes Larger.

→ Each symbol of the message reduces the size of the interval in accordance with its probability of occurance.

Encoding Sequence ⟶

a_1 a_2 a_3 a_3 a_4



Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

The message is assumed to occupy the entire half open interval [0,1). The interval is subdivided into four regions based on the probabilities of each source symbol.
Symbol a_1 is associated with subinterval [0,0.2). Because it is the first symbol of the message being coded is narrowed to [0,0.2).
The narrowed range is then subdivided in accordance with the original source symbol probabilities and the process continues with the next message symbol.
→ In this manner symbol a_2 narrows to sub interval [0.04, 0.08)
a_3 further narrows to [0.056, 0.072) and so on.
→ The final message symbol narrows to [0.06752, 0.0688)
→ The 0.068 - can be used to represent the message.

LZW Coding :-

LZW coding is one of the several error-free compression techniques that also attack an images interpixel redundancies.
→ Lempel-Ziv-Welch (LZW) coding assigns fixed-length code words to variable length sequences of source symbols
→ LZW compression has been integrated into a variety of mainstream imaging file formats including the graphic interchange format (GIF), tagged image file format (TIFF) and the portable document format (PDF).
→ At first onset of coding process. a code book or dictionary containing the source symbols to be coded is constructed.
→ For 8-bit monochrome images the first 256 words of the dictionary are assigned to the gray values 0,1,2 ----255

As the encoder sequentially examines the images pixels gray level sequences that are not in the dictionary are placed into algorithmically determined Locations

→ If first two pixels are white for instance sequence "255-255" might be assigned to Location 256.

→ The next time that two consecutive white pixels are encountered code word 256 is used to represent them.

<u>Example:-</u> 4x4, 8-bit image of a vertical edge.

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Locations 256 through 511 are initially unused.

LZW Coding Example

Currently Recognized sequence	Pixel Being processed	Encoded Output	Dictionary Location (code word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126	258	261	126-126-39
126-126	39			
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
36	126			
36-126	126	257	264	39-126-126

A unique feature of the LZW coding is that the coding dictionary or code book is created while the data are being encoded.

→ LZW decoder builds an identical decompression dictionary as it decodes simultaneously the encoded data stream.

Bit-Plane Coding :-

Bit-plane coding is an effective technique for reducing an images interpixel redundancies by processing the image's bit planes individually.

→ The method is based on the concept of decomposing a multilevel images into a series of binary images and compressing each binary image via one of several well-known binary compression methods.

Bit-plane decomposition :-

The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0$$

→ A simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit planes.

→ Zeroth-order bitplane is generated by collecting all a_0 bits of each pixels.

→ while the $(m-1)^{th}$ order bit plane contains the a_{m-1} bits.

→ The inherent disadvantage of this approach is that small changes in gray level can have a significant impact on the complexity of the ~~bit~~ bit planes.

→ If a pixel of intensity 127 (01111111) is adjacent to a pixel of intensity 128 (10000000) every bit pl is manipulated.

An alternative decomposition approach is to first represent the image by an m-bit Gray code.

→ The m-bit Gray code $g_{m-1} \cdots g_2 g_1 g_0$ that is computed as

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2$$

$$g_{m-1} = a_{m-1}$$

$\oplus$ denotes exclusive OR operation.

This code has the unique property that successive code words differ in only one bit position.

→ Gray codes for 127 and 128 are 11000000 and 01000000 respectively.

Lossless Predictive Coding :-

An error-free compression approach that does not require decomposition of an image into a collection of bit planes, called as Lossless predictive coding.

→ it is based on eliminating the interpixel redundancies of closely spaced pixels by extracting and coding only the new information in each pixel.

→ The new information of a pixel is defined as the difference between the actual and predicted value of the pixel.

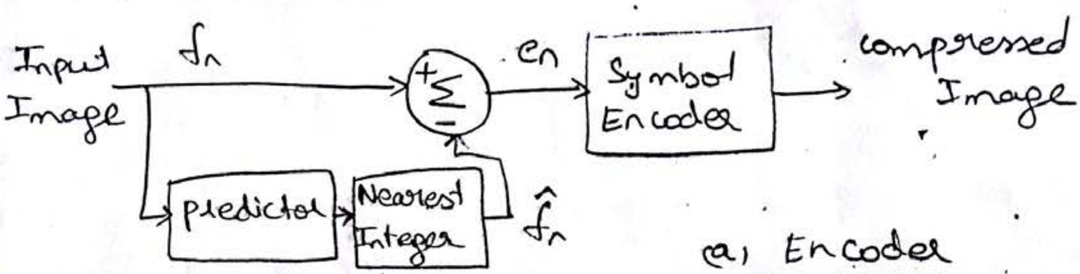
→ The Lossless predictive coding system consists of an encoder and a decoder each containing an identical predictor.

→ f_n denote successive pixel of the input image given to predictor which generates the anticipated value of that pixel based on some number of past inputs.

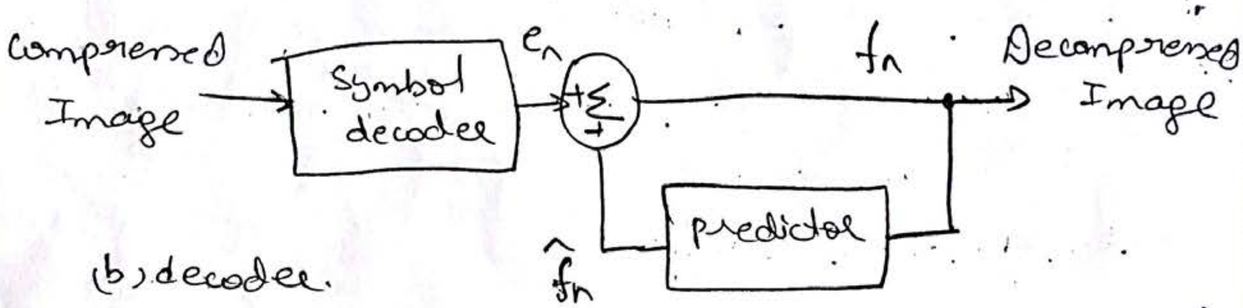
→ The output of the predictor is then rounded to the nearest integer denoted $\hat{f}_n$.

The difference of prediction error is given by below eq and is coded with variable-length code.

$$e_n = f_n - \hat{f}_n$$



(a) Encoder



(b) decoder.

Fig: A lossless predictive coding Model.

→ The decoder reconstructs e_n from the received variable length code words and performs the inverse operation

$$f_n = e_n + \hat{f}_n$$

Various local, global and adaptive methods can be used to generate $\hat{f}_n$.

→ The prediction is formed by a linear combination of m previous pixels.

$$\hat{f}_n = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

where m is the order of the linear predictor

→ round is a function for nearest integer operation.

α_i for $i = 1, 2, \ldots m$ are prediction coefficients.

→ $$\hat{f}_n(x,y) = \text{round}\left[\sum_{i=1}^{m} \alpha_i f(x, y-i)\right]$$

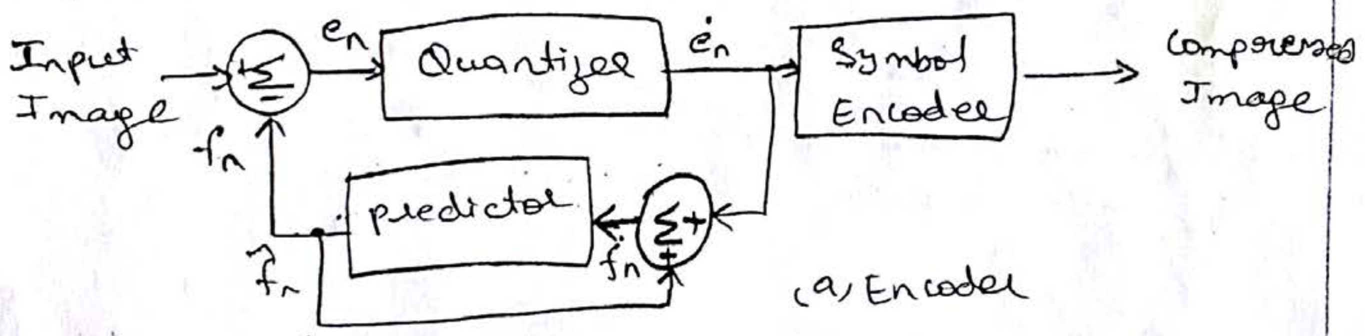
7) Lossy Compression :-

Lossy Encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression.

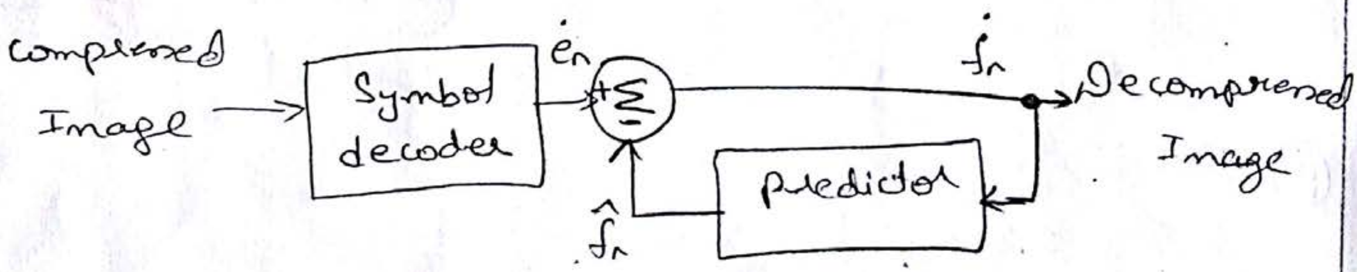
Lossy predictive coding :-

Lossy predictive coding is similar to lossless predictive coding except we add a quantizer to the model.

→ The Quantizer absorbs the nearest integer function of the error-free encoder and is inserted between the symbol encoder and the point at which the prediction error is formed.



(a) Encoder



b) decoder

fig: A Lossy predictive coding model

→ $\dot{e}_n$ denote the amount of compression and distortion associated with Lossy predictive coding.

→ $\dot{f}_n$ denote the input to lossy encoder's predictor within a feedback loop.

$$\boxed{\dot{f}_n = \dot{e}_n + \hat{f}_n}$$

→ Delta Modulation (DM) is a simple but well-known form of lossy predictive coding in which the predictor and quantizer are defined as

$$\boxed{\hat{f}_n = \alpha \dot{f}_{n-1}}$$

and
$$\dot{e}_n = \begin{cases} +\zeta & \text{for } e_n > 0 \\ -\zeta & \text{otherwise.} \end{cases}$$

where α is a prediction coefficient
ζ is a positive constant.

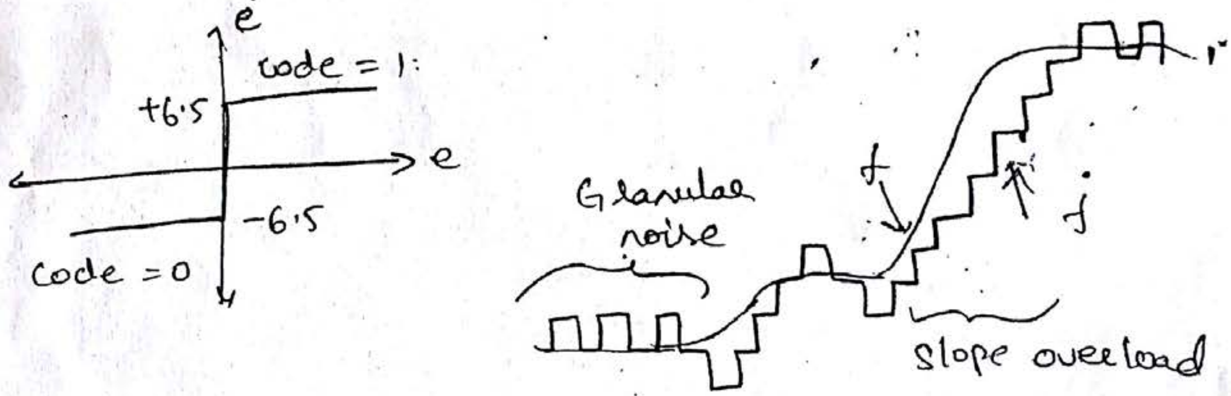


Fig:- An Example of delta modulation.

(8) JPEG (Joint

One of the most popular and comprehensive continuous tone, still frame compression standards is the JPEG Standard.

It defines three different coding systems.

(1) a lossy baseline coding system, which is based on the DCT and is adequate for most compression applications

(2) an extended coding system for greater compression, higher precision or progressive reconstruction applications

(3) a lossless independent coding system for reversible compression.

→ The compression is performed in three sequential steps:

(1) DCT computation

(2) Quantization

(3) Variable-length code assignment.

→ The image is first subdivided into pixel blocks of size 8×8, which are processed left to right top to bottom.

→ 8×8 block or subimage is encountered, its 64 pixels are level shifted by subtracting the quantity 2^{n-1} where 2^n is the maximum number of gray levels.

→ ~~One-dimension~~ The 2-D discrete cosine transform of the block is then computed, quantized ~~in accordance~~ and reordered using the zigzag pattern to form a 1-D sequence of quantized coefficients.