Image Processing and Computer Vision

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

RUN LENGTH CODING

- Run-length coding is a very widely used and simple compression technique which does not assume a memoryless source
  - We replace runs of symbols (possibly of length one) with pairs of (*run-length*, *symbol*)
  - For images, the maximum run-length is the size of a row
- Run length coding (RLC) is effective when long sequences of the same symbol occur.
- Run length coding exploits the spatial redundancy by coding the number of symbols in a run.

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

RUN LENGTH CODING

- The term run is used to indicate the repetition of a Symbol, while the term run length is used to represent the number of repeated symbols.
- Run length coding maps a sequence of numbers into a sequence of symbol pairs (run value).
- Images with large areas of constant shade are good candidates for this kind of compression.
- It is used in windows bitmap file format.
- Run length coding can be classified into

I) 1-D run length coding.

II) 2-D run length coding.

RUN LENGTH CODING

I) I-D run length coding-

In 1-D run length coding, each scan line is encoded independently.

1-D RLC utilizes only the horizontal correlationship between pixels on the same scan line.

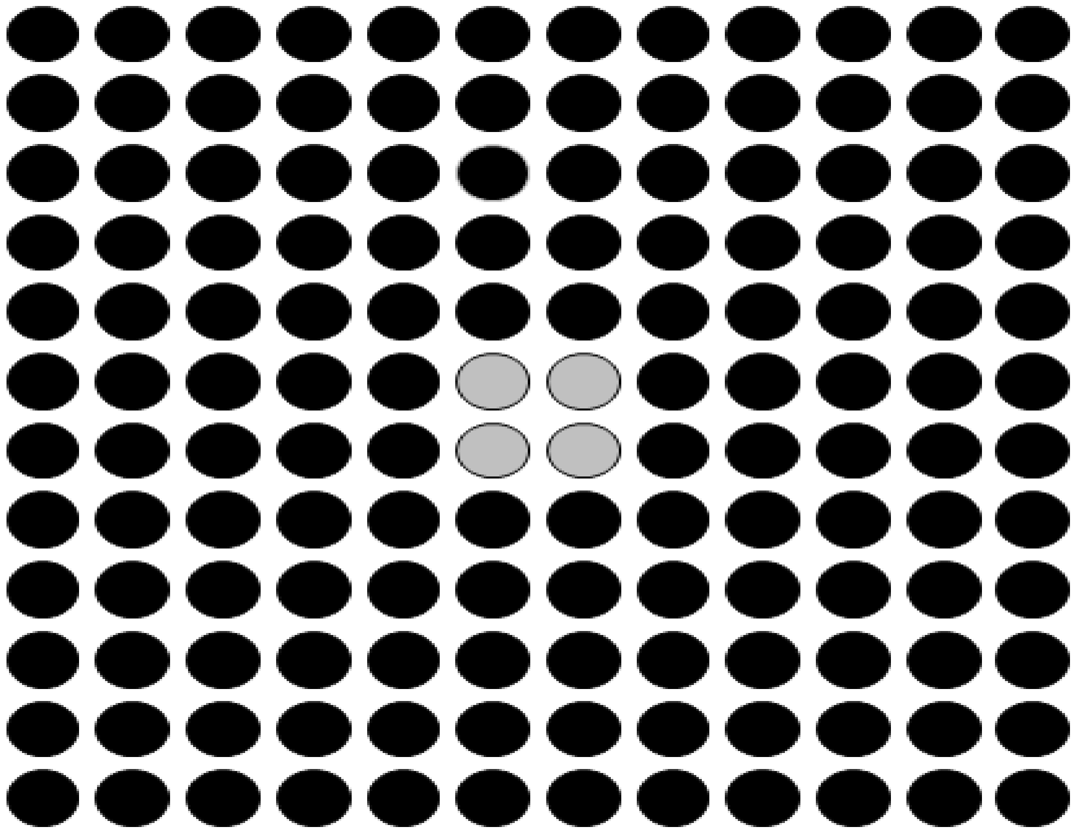
2-D RLC utilizes both horizontal and vertical correlationship between pixels.

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

RUN LENGTH CODING

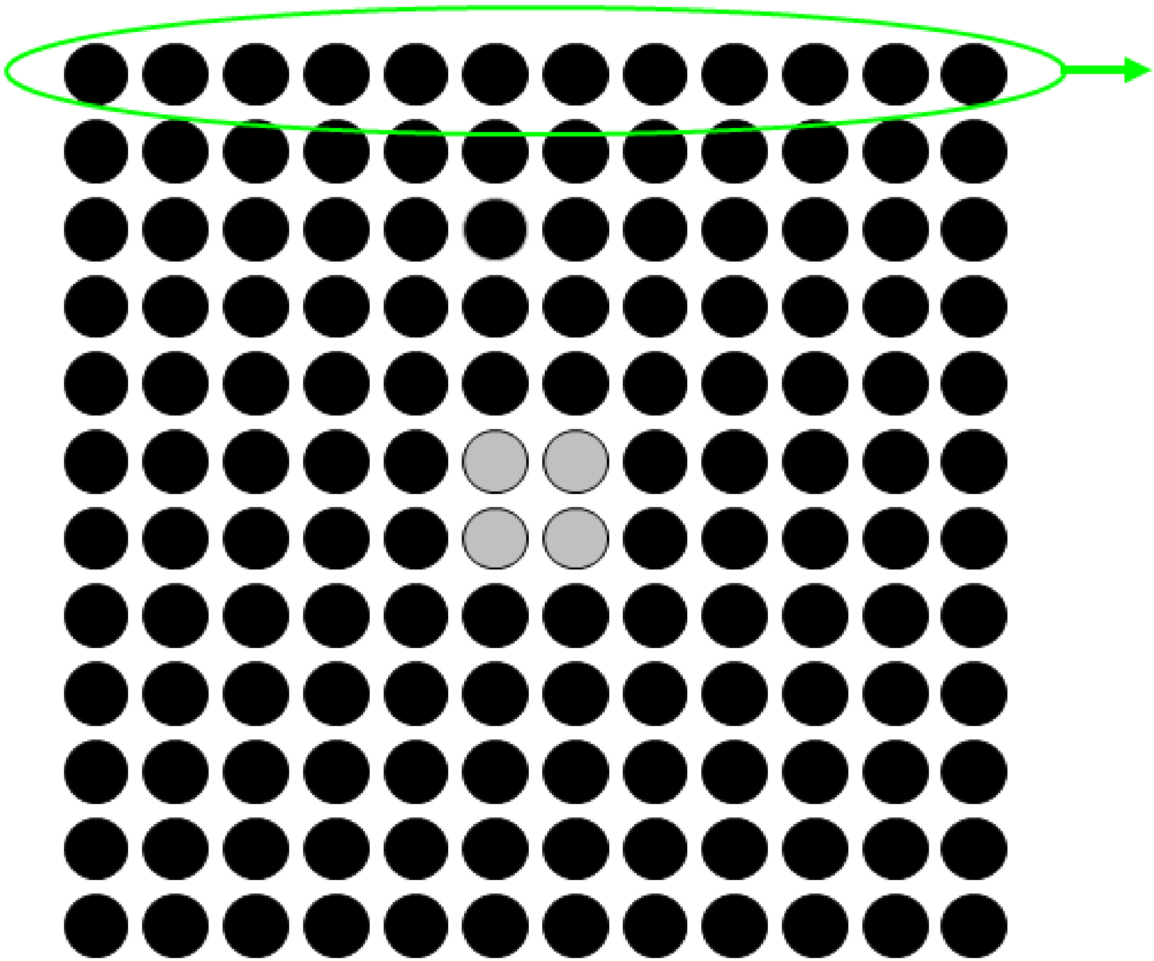
- 2-D run length coding
- The 1-D run length coding utilizes the correlation between pixels with in a scanline.
- In order to utilize correlation between pixels in neighbouring scan lines to achieve higher coding efficiency, 2-D run length coding was developed.
- In RLC, two values are transmitted- the first value indicates the number of times a particular symbol has occurred. The second value indicates the actual symbol.

RUN LENGTH CODING



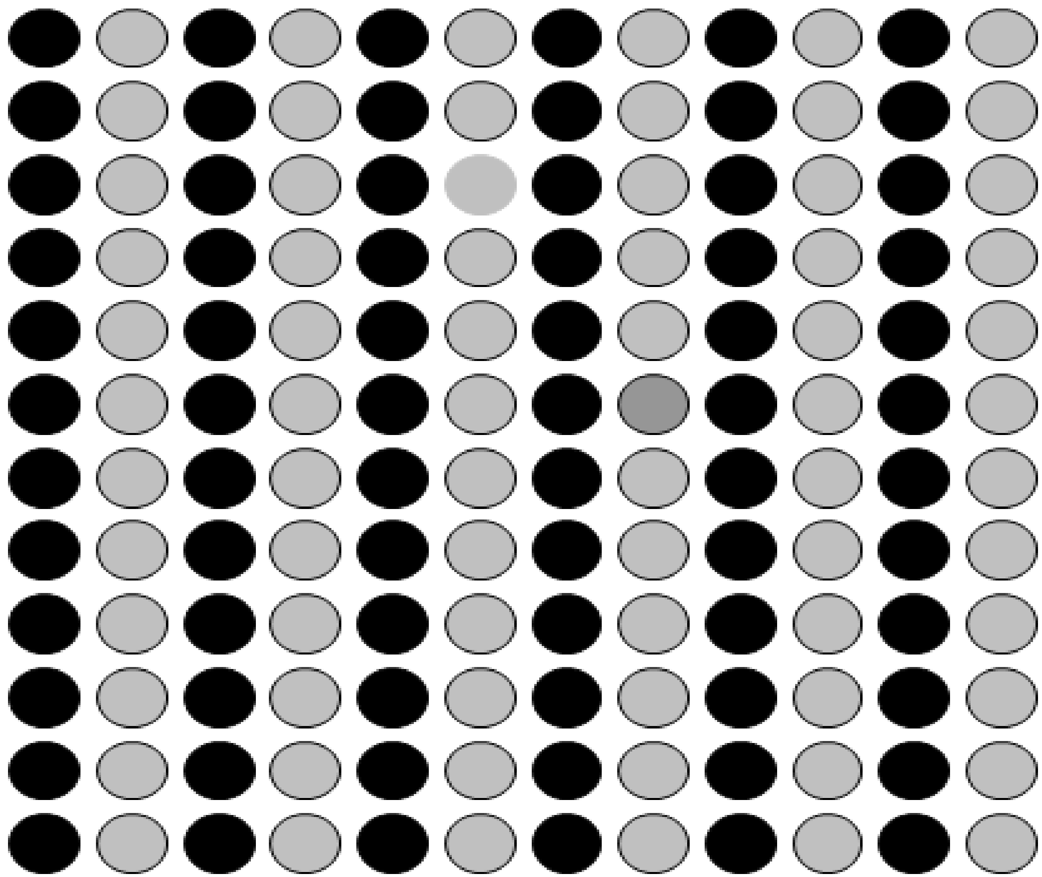
Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

RUN LENGTH CODING



0(12)
0(12)
0(12)
0(12)
0(12)
0(5)1(2)0(5)
0(5)1(2)0(5)
0(12)
0(12)
0(12)
0(12)
0(12)

RUN LENGTH LIMITATIONS



Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

ENTROPY

- The average bit rate of a coding scheme is given by:

$$\bar{b} = \sum_{i=1}^{L} \mathrm{p}_i \, b_i$$

- p_i = probability of occurrence of the i^{th} reconstruction level
- b_i = number of bits assigned to the i^{th} message

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

ENTROPY

- In information theory, **entropy** is a measure of the uncertainty associated with a random variable.
- This term usually refers to the **Shannon entropy**, which quantifies the expected value of the information contained in a message.
- Entropy is typically measured in bits.
- Entropy, in an information sense, is a measure of unpredictability.
- Compressed message is more unpredictable, because there is no redundancy; each bit of the message is communicating a unique bit of information.
- Entropy is a measure of how much information the message contains.

ENTROPY

E = entropy(I)

E=a scalar value representing the entropy of grayscale image I. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = \sum_{i=1}^{n} p(x_i) \log_b \frac{1}{p(x_i)} = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i),$$

where p contains the histogram counts returned from imhist.
where b is the base of the logarithm used. Common values of b are 2,10.

A SIMPLE EXAMPLE

- Suppose we have a message consisting of 5 symbols, e.g. [►♣♣♠☻►♣☼► ☻]
- How can we code this message using 0/1 so the coded message will have minimum length (for transmission or saving!)

►	000
♣	001
☻	010
♠	011
☼	100

- 5 symbols → at least 3 bits
- For a simple encoding,

  length of code is 10*3=30 bits

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

A SIMPLE EXAMPLE – CONT.

- Intuition: Those symbols that are more frequent should have smaller codes, yet since their length is not the same, there must be a way of distinguishing each code

- For Huffman code,

length of encoded message

will be ►♣♣♠☻►♣☼►☻

=3*2 +3*2+2*2+3+3=24bits

Symbol	Freq.	Code
►	3	00
♣	3	01
☻	2	10
♠	1	110
☼	1	111

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

ENTROPY CODING

Example 7.3 Calculate the entropy for the symbols shown in Table 7.2.

Table 7.2 Symbols and their distribution

Symbol	1	2	3	4	5	6
Probability	0.4	0.2	0.2	0.1	0.05	0.05

Solution Entropy $= -\Sigma p_i \times \log_2 p_i$; as $\log_2 x = \log_{10} x / \log_{10} 2$

$= -[0.4 \times (\log10(0.4)/\log10(2)) + 0.2 \times (\log10(0.2)/\log10(2)) + 0.2 \times (\log10(0.2)/\log10(2)) + 0.1 \times (\log10(0.1)/\log10(2)) + 0.05 \times (\log10(0.05)/\log10(2)) + 0.05 \times (\log10(0.05)/\log10(2))]$

$= -[-0.5288 - 0.4644 - 0.4644 - 0.3322 - 0.2161 - 0.2161]$

$= 2.22$

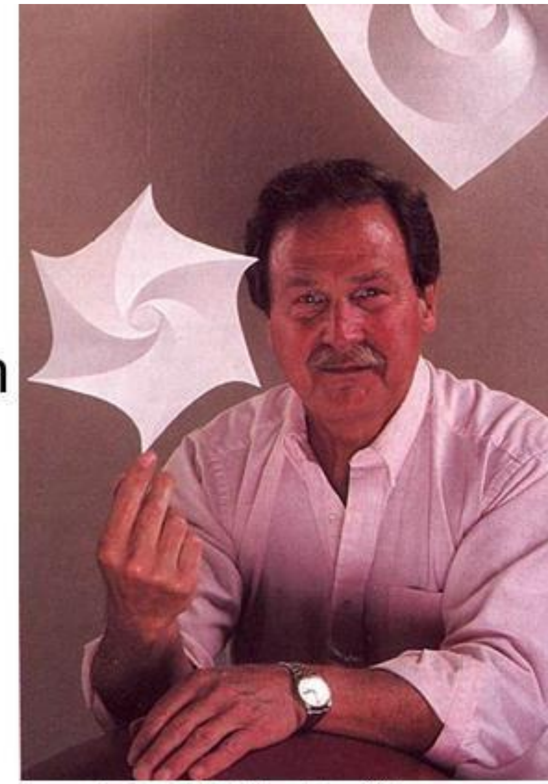
Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HUFFMAN CODE

- Developed by David A. Huffman in 1951
- Widely used in Computers, HDTV, modems etc
- Robert M. Fano assigned a term paper on efficient representation of numbers

Robert M. Fano

Claude Shannon



David A. Huffman

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HUFFMAN CODING

- Huffman codes are optimal codes that map one symbol with one code word.
- In Huffman coding, it is assumed that each pixel intensity has associated with it a certain probability of occurrence, and this probability is spatially invariant.
- Huffman coding assigns a binary code to each intensity value, with shorter codes going to intensities with higher probability.
- If the probabilities can be estimated then the table of Huffman codes can be fixed in both the encoder and the decoder.

HUFFMAN CODING

- The Parameters involved in Huffman coding are as follows.
- **Entropy**
- **Average Length**
- **Efficiency**
- **Variance.**

- **Prefix Code-**A code is a prefix code if no code word is the prefix of another code word. The main advantage of a prefix code is that it is uniquely decodable.

  Example-Huffman code.

HUFFMAN CODING ALGORITHM

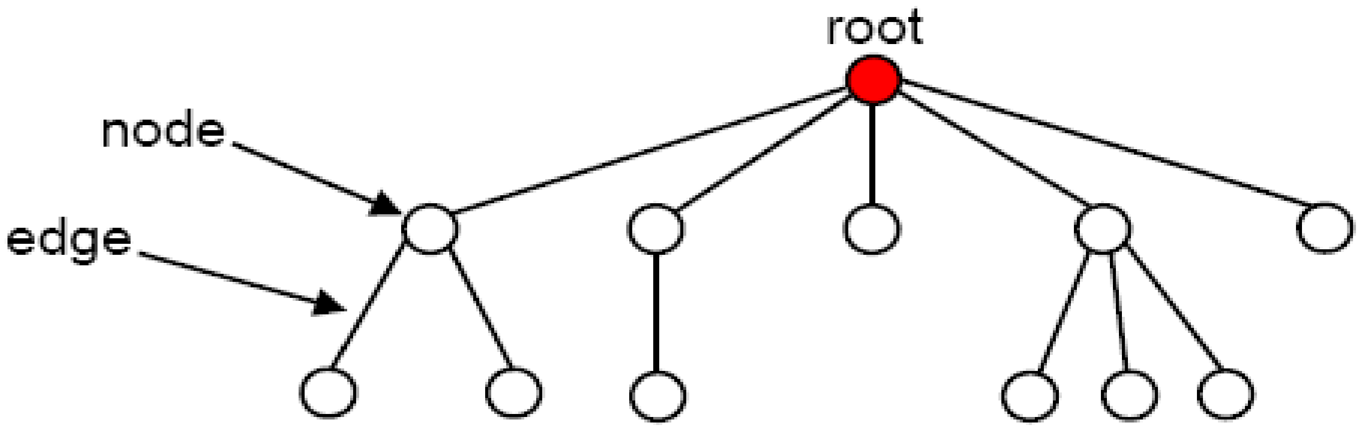
- Initialization: Put all symbols on a list sorted according to their frequency counts.
- From the list pick two symbols with the lowest frequency counts. Form a Huffman subtree that has these two symbols as child nodes and create a parent node.
- Assign the sum of the children's frequency counts to the parent and insert it into the list such that the order is maintained.
- Delete the children from the list.
- Repeat until the list has only one symbol left
- Assign a codeword for each leaf based on the path from the root.

PROPERTIES OF HUFFMAN CODES

- No Huffman code is the prefix of any other Huffman codes so decoding is unambiguous
- The Huffman coding technique is optimal (but we must know the probabilities of each symbol for this to be true)
- Symbols that occur more frequently have shorter Huffman codes

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

WHAT IS A TREE?



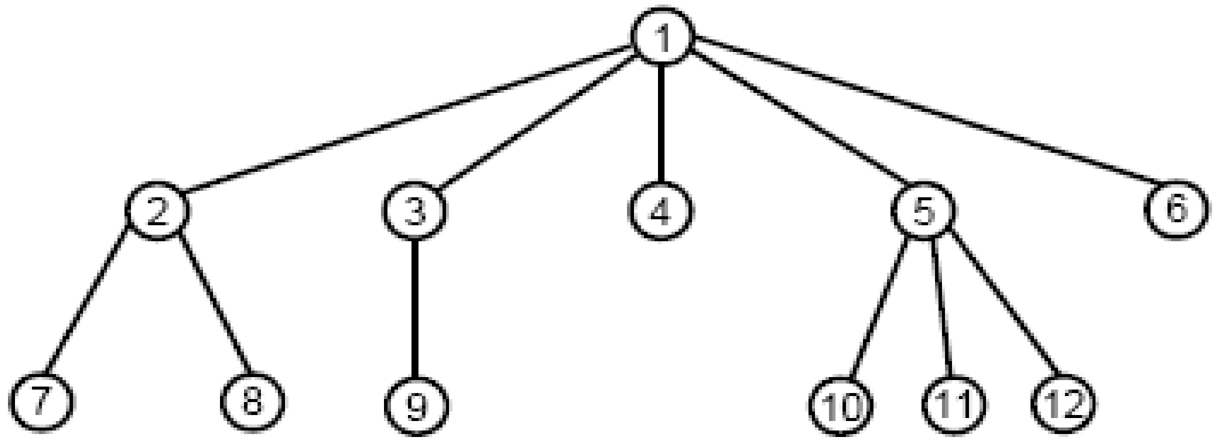
A tree consists of:

- a set of *nodes*
- a set of *edges*, each of which connects a pair of nodes.

The node at the "top" of the tree is called the *root* of the tree.

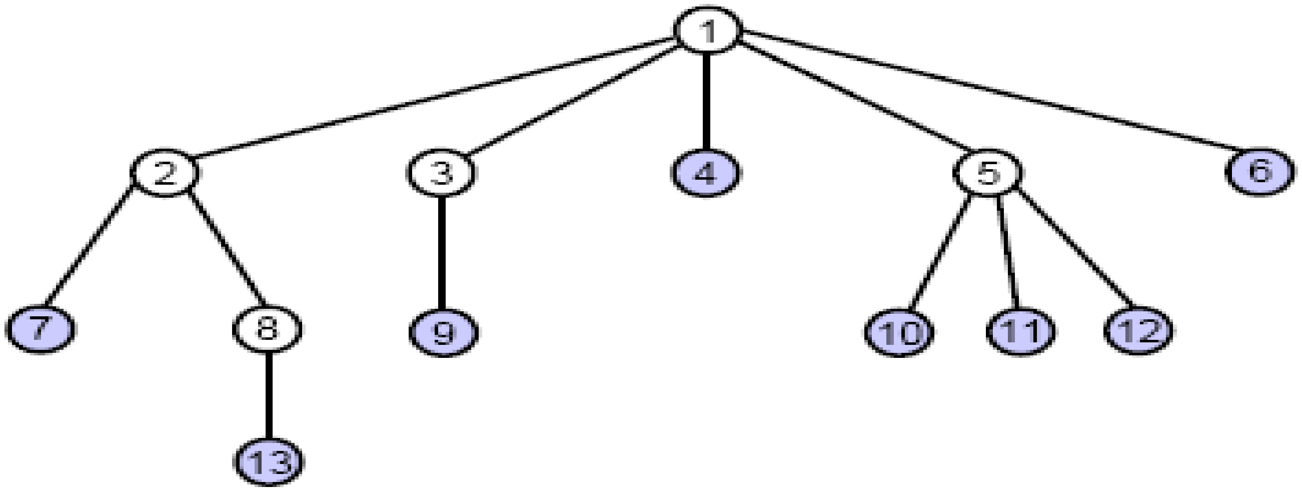
Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

RELATIONSHIPS BETWEEN NODES



- If a node N is connected to other nodes that are directly below it in the tree, N is referred to as their *parent* and they are referred to as its *children*.
- example: node 5 is the parent of nodes 10, 11, and 12
- Each node is the child of *at most one* parent.

TYPES OF NODES



- A *leaf node* is a node without children.
- An *interior node* is a node with one or more children.

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

EXAMPLE

Example 9.2 *Obtain the Huffman code for the word 'COMMITTEE'*

Solution Total number of symbols in the word **'COMMITTEE'** is 9.

$$\text{Probability of a symbol} = \frac{\text{Total number of occurrence of symbol in a message}}{\text{Total number of symbol in the message}}$$

Probability of the symbol C = $p(\text{C}) = 1/9$

Probability of the symbol O = $p(\text{O}) = 1/9$

Probability of the symbol M = $p(\text{M}) = 2/9$

Probability of the symbol I = $p(\text{I}) = 1/9$

Probability of the symbol T = $p(\text{T}) = 2/9$

Probability of the symbol E = $p(\text{E}) = 2/9$

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

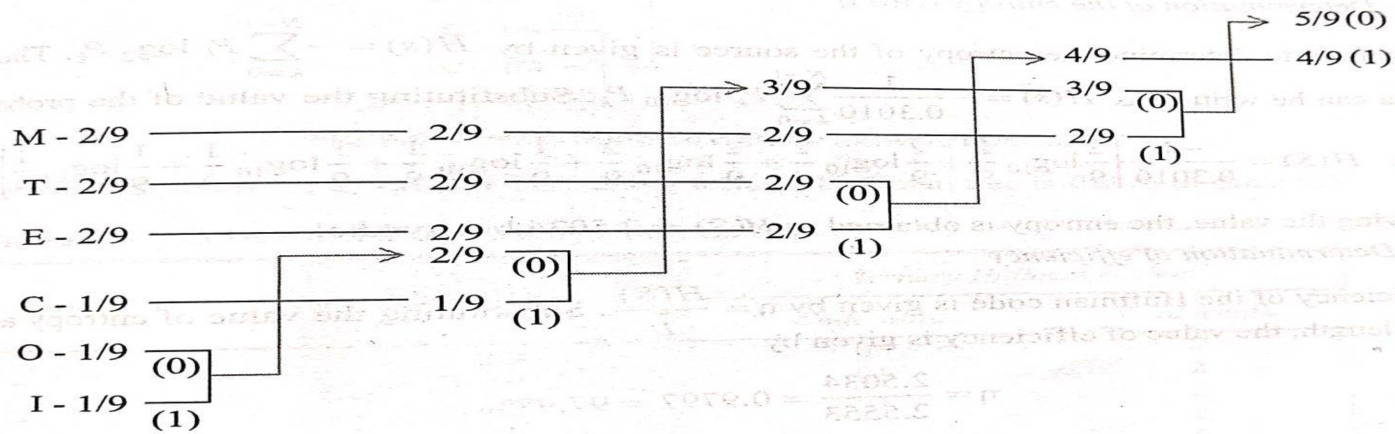
Step 1 Arrange the symbols into descending order according to the probability. This is illustrated in Table 9.1.

Table 9.1 *Arrangement of symbols in descending order of probability*

Symbol	*Probability*
M	2/9
T	2/9
E	2/9
C	1/9
O	1/9
I	1/9

Step 2 Construction of Huffman tree

The Huffman tree corresponding to the term COMMITTEE is shown below.



Step 3 Code word from the Huffman Tree

The code word for each symbol and the corresponding word length is shown in Table 9.2.

Table 9.2 *Code word from Huffman tree*

Symbol	*Probability*	*Binary Huffman method*	
		Codeword	*Word length*
M	2/9	01	2
T	2/9	10	2
E	2/9	11	2
C	1/9	001	3
O	1/9	0000	4
I	1/9	0001	4

$$H(S) = \frac{-1}{0.3010}\left\{\frac{2}{9}\log_{10}\frac{2}{9} + \frac{2}{9}\log_{10}\frac{2}{9} + \frac{2}{9}\log_{10}\frac{2}{9} + \frac{1}{9}\log_{10}\frac{1}{9} + \frac{1}{9}\log_{10}\frac{1}{9} + \frac{1}{9}\log_{10}\frac{1}{9}\right\}$$

$$H(S) = 2.5034 \text{ bits/symbol}$$

Finding the average length $\overline{L}$ (Ternary Huffman Coding)

$$\overline{L} = \sum_{k=0}^{N-1} P_k l_k = \frac{2}{9}\times 2 + \frac{2}{9}\times 2 + \frac{2}{9}\times 2 + \frac{1}{9}\times 2 + \frac{1}{9}\times 2 + \frac{1}{9}\times 2$$

$$= 0.4444 + 0.4444 + 0.4444 + 0.2222 + 0.2222 + 0.2222$$

$$\overline{L} = 1.9998 \text{ bits/symbol}$$

***Efficiency of the compression in ternary Huffman Coding* (η)**

The efficiency of the Huffman code is given by

$$\eta = \frac{H(S)}{\overline{L}}$$

By substituting the value of $H(S)$, $\overline{L}$ we get

$$\eta = \frac{2.5034}{1.9998} = 125.18\%$$

$$\eta = 1.2518$$

HUFFMAN CODING EXAMPLE

Input Image

$$\begin{bmatrix} 1 & 2 & 5 & 7 \\ 2 & 3 & 7 & 5 \\ 7 & 2 & 1 & 3 \\ 6 & 4 & 7 & 1 \end{bmatrix}$$

STEP: 1

Probability of occurrence of gray level

Probability of '1' = P(1) = 3/16

Probability of '2' = P(2) = 3/16

Probability of '3' = P(3) = 2/16

Probability of '4' = P(4) = 1/16

Probability of '5' = P(5) = 2/16

Probability of '6' = P(6) = 1/16

Probability of '7' = P(7) = 4/16

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HUFFMAN CODING EXAMPLE

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
	7	4/16					
							12

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HUFFMAN CODING EXAMPLE

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
	7	4/16					
	1	3/16					
	2	3/16					
	3	2/16					
	5	2/16					
	4	1/16					
	6	1/16					12

HUFFMAN CODING EXAMPLE

Pixel Value	1	2	3	4	5	6	7
Probability of occurrence	3/16	3/16	2/16	1/16	2/16	1/16	4/16

STEP 2: **Arrange the probability in descending order**

Code	Symbol	Probability	Step 1	Step 2	Step 3	Step 4	Step 5
(01)	7	4/16	4/16	4/16	5/16	7/16	9/16 (0)
(11)	1	3/16	3/16	4/16	4/16	5/16 (0)	7/16 (1)
(000)	2	3/16	3/16	3/16	4/16 (0)	4/16 (1)	
(001)	3	2/16	2/16	3/16 (0)	3/16 (1)		
(100)	5	2/16	2/16 (0)	2/16 (1)			
(1010)	4	1/16 (0)	2/16 (1)				
(1011)	6	1/16 (1)					12

HUFFMAN CODING EXAMPLE

Pixel Value (or) Symbol	1	2	3	4	5	6	7
Probability, P_k	3/16	3/16	2/16	1/16	2/16	1/16	4/16
Code	11	000	001	1010	100	1011	01

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune

HUFFMAN CODING EXAMPLE

STEP: 3 **To find the Average length**

$$\bar{L} = P_1 l_1 + P_2 l_2 + P_3 l_3 + P_4 l_4 + P_5 l_5 + P_6 l_6 + P_7 l_7$$

$$= \frac{3}{16}\times 2 + \frac{3}{16}\times 3 + \frac{2}{16}\times 3 + \frac{1}{16}\times 4 + \frac{2}{16}\times 3 + \frac{1}{16}\times 4 + \frac{4}{16}\times 2$$

$$= \frac{1}{16}[6 + 9 + 6 + 4 + 6 + 4 + 8]$$

$$= \frac{43}{16} = 2.6875$$

Where P_k is Probability of the symbol or pixel value

l_k is length of the code for the corresponding symbol

$\bar{L}$ is Average length

STEP: 4 To find the Entropy, H(s)

$$H(s) = -\sum_k P_k \log_2^{(P_k)}$$

$$H(s) = -\left\{ \frac{3}{16} \times \log_2^{\left(\frac{3}{16}\right)} + \frac{3}{16} \times \log_2^{\left(\frac{3}{16}\right)} + \frac{2}{16} \times \log_2^{\left(\frac{2}{16}\right)} + \frac{1}{16} \times \log_2^{\left(\frac{1}{16}\right)} + \frac{2}{16} \times \log_2^{\left(\frac{2}{16}\right)} + \frac{1}{16} \times \log_2^{\left(\frac{1}{16}\right)} + \frac{4}{16} \times \log_2^{\left(\frac{4}{16}\right)} \right\}$$

$$= -\left\{ \frac{3}{16} \times (-2.4153) + \frac{3}{16} \times (-2.4153) + \frac{2}{16} \times (-3.0003) + \frac{1}{16} \times (-4.0004) + \frac{2}{16} \times (-3.0003) + \frac{1}{16} \times (-4.0004) + \frac{4}{16} \times (-2) \right\}$$

$$= \frac{1}{16}\{7.2459 + 7.2459 + 6.0006 + 4.0004 + 6.0006 + 4.0004 + 8\}$$

$$= \frac{1}{16} \times 42.4938 = 2.6559$$

STEP: 5 To find the Efficiency $\eta = \frac{H(s)}{\bar{L}}$

$$\eta = \frac{2.6559}{2.6875}$$

$$= 0.9882$$

Efficiency in Percentage =0.9882 x 100 = **98.82**%

Prof.(Dr.) Deepali S. Jadhav
Dept. of Information Technology, VIT, Pune