

Unit - III

Huffman Coding. (Introduced by Huffman - 1952).

- A measure to reduce ^{removing} coding redundancy.
- Most popular coding redundancy technique.
- Variable length code.

Example

Image size 10×10 .

Frequency

a1	- 10
a2	- 40
a3	- 6
a4	- 10
a5	- 4
a6	- 30

Average loading.

a2 = 40
a6 - 30
a1 - 10
a4 - 10
a3 - 6
a5 - 4

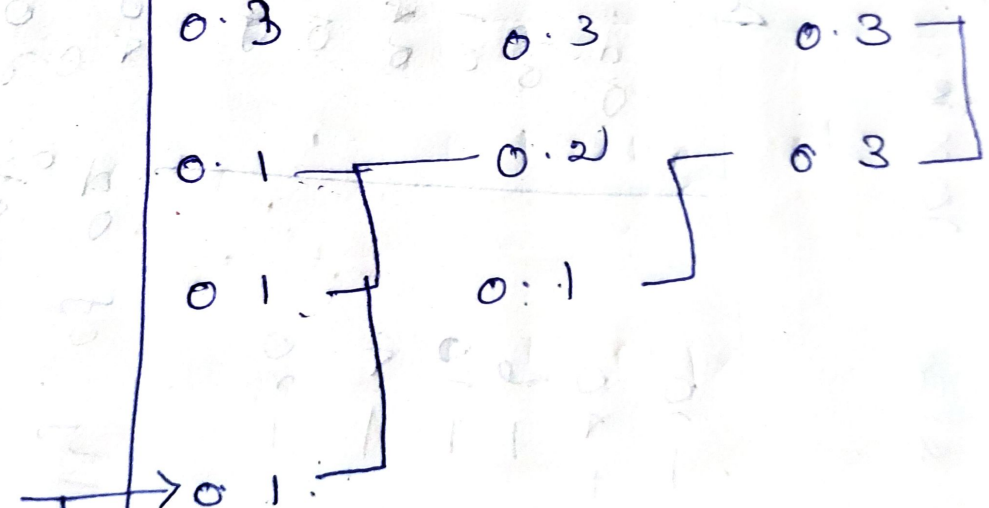
Probability of a2

$$\{a1, a2, a3, a4, a5, a6\} = \{0.1, 0.4, 0.06, 0.1, 0.04, 0.3\}$$

$$40/100 = 0.4$$

Huffman Source Reduction.

Original Source		Source Reduction.			
Symbol.	Probability	1	2	3	4.
a ₂	0.4	0.4	0.4	0.4	0.4
a ₆	0.3	0.3	0.3	0.3	0.4
a ₁	0.1	0.1	0.2	0.3	
a ₄	0.1	0.1	0.1		
a ₅	0.06	0.1			
a ₆	0.04				



Step 1:

- Create a series of source reductions by ordering the probabilities of symbols.
- (Symbol - intensities of image)
- First source reduction is formed by combining 0.06 and 0.04
 $0.06 + 0.04 = 0.1$.
- In the source reduction, I column, write this 0.1 instead of 0.06 & 0.04.
- Second reduction ($0.1 + 0.1 = 0.2$) -
0.2 is written as a 3rd number in II column
- In column III we get only 2 values so we can stop the reduction.

Step 2: Huffman Code Assignment Procedure

- In this step, each reduced source is coded.
- It starts from the smallest source obtained in the last step and goes back to the original source.
- The minimal length binary codes are used are 0 and 1.

- The reduced symbols 0.6 and 0.4 in the last column are assigned 0 and 1.
- Since 0.6 was generated ^{by combining} 0.3 + 0.3.
So symbols in the reduced source to its part so 0 and 1 are appended which produces codes 00 and 01.
- 0.3 was generated by 0.2 + 0.1 -
Codes - 010 and 011 codes produced.
- This operation is repeated for each produced source until the original source is reached.
- The average length of this code is

$$L_{avg} = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + 0.04(5)$$

$$= 2.2 \text{ bits/pixel} \dots$$

Advantages:

- It creates an optimal code for a set of symbols and probability.
- Implementation is very simple.

Drawbacks:

- It is difficult to implement when a large no. of symbols has to be coded.