MEDICAL IMAGING PROCESSING

FOURTH STAGE

2025-2026

Image Compression

BY

MS.c Mortada Sabri **MS.c Najwan Thaeer Ali**

Lec 6

Outline

- Introduction to Image/Text Compression
- Coding Redundancy
- Steps of Huffman Encoding
- Huffman Encoding Example
- Advantages of Huffman Coding

Introduction

Data compression is an important aspect of contemporary digital systems that require storing and transmitting large volumes of images, text, and multimedia. **Huffman** encoding is one of the most popular ways of lossless compression and is a technique used to assign symbols with varying lengths of binary codes based on their occurrence rate.

Symbols with higher frequency are assigned shorter codes, and those least frequent are assigned longer ones, so the total number of bits needed to represent the information is less. Since Huffman coding does not lose data, one can perfectly reassemble the original data; therefore, it is very useful in image compression, file compression (ZIP/GZIP), and text encoding.

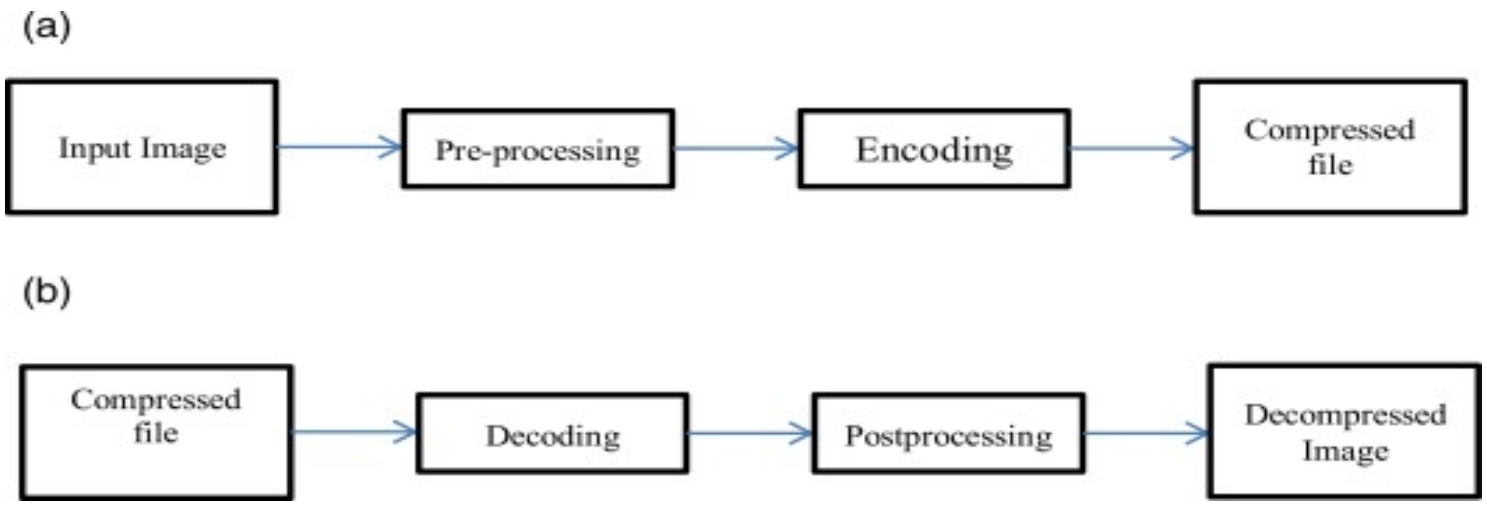


Image Compression

CODING: Fewer bits to represent frequent symbols.

COMPRESSION: The process of coding that reduces the total number of bits needed to represent information.

Objective: reduce the amount of data required to represent an image.

There are two main categories:

Lossless compression: All information is retained. Suitable for computer programs, medical images, GIS, .gif files.

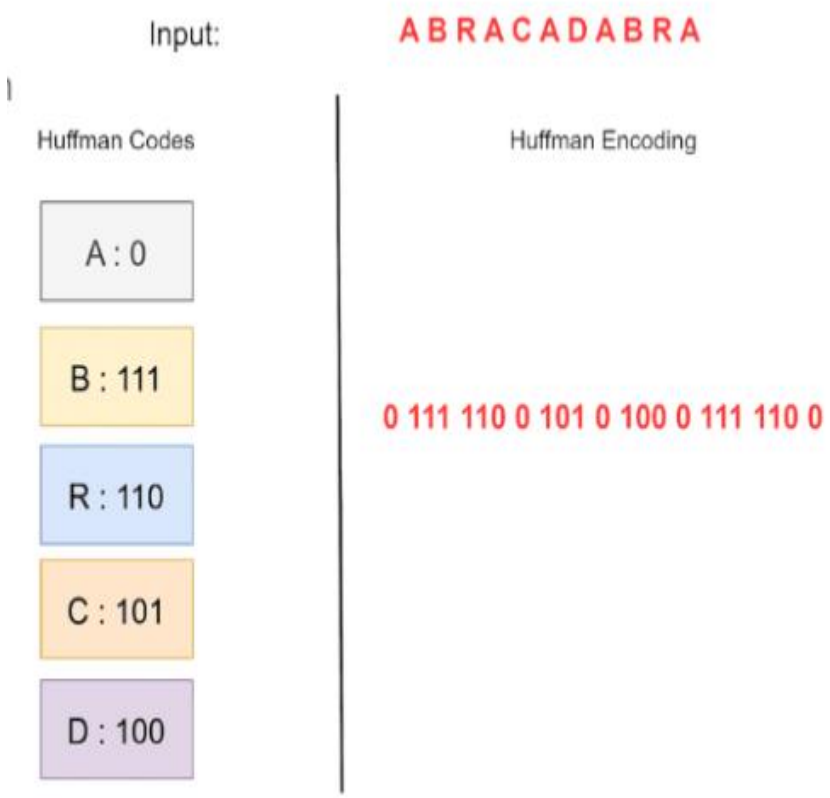
Loss compression: Some information is lost. Used in TV signals, teleconferencing, .mp3,jpg.

Image Compression

General Data Compression Scheme:

Input data → Encoder (compression) → Storage/Network → Decoder (decompression) → Output data.

Compression ratio = (B_0 / B_1)
Where:
(B_0) = bits before compression
(B_1) = bits after compression



Frequency Analysis

Frequency Analysis First, we analyze the frequency of each character (or symbol) in the input data. This involves counting how many times each character appears in the data (Figure 2). This frequency analysis helps determine which characters occur most and least frequently.

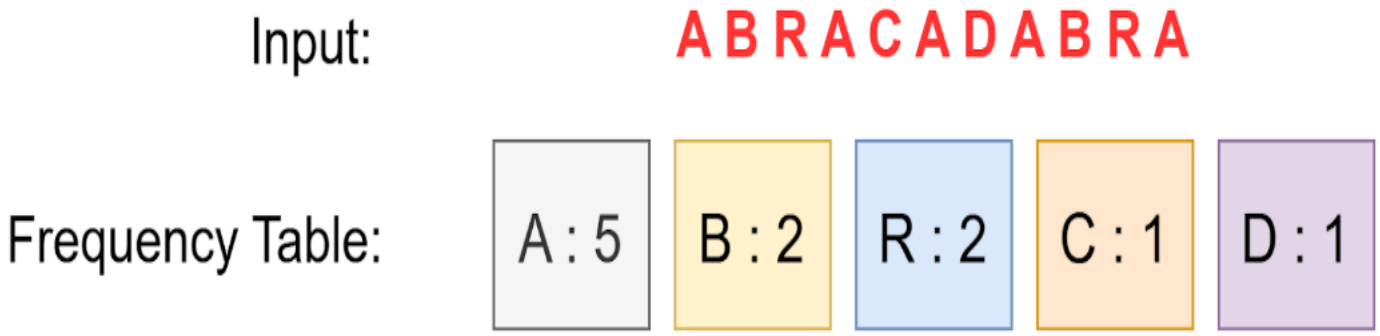


Figure 2: Frequency Analysis Table (source: image by the author).

Run-Length Encoding (RLE)

RLE is a simple form of compression where **runs of data** (repeated values) are stored as a single value and count.

Useful for data containing many runs (icons, line drawings, animations).
Not useful for data without many runs.

Example 1

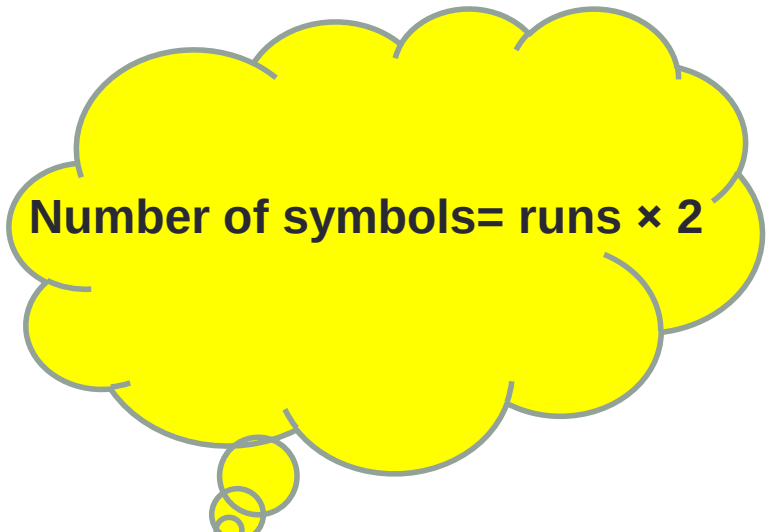
Screen of black text on white background: long runs of white pixels, short runs of black pixels. Hypothetical scan line (W=white, B=black):

WWWWWWWWWWWWWWBBBBWWWWWWWWWWBBBWWWWWWWWWWW

Run-length encoding result:

13W 4B 8W 3B 11W

Run-length code uses **10 symbols instead of 39**.



Fixed-Length vs Variable-Length Codes

Example 2

Characters: a, b, c, d, e, f

Frequencies: 45, 13, 12, 16, 9, 5

$$6 > 4 = {}^{2}2$$

$$6 \leq 8 = {}^{3}2$$

Sol //.

We have 6 symbols → How many bits do we need to represent them?

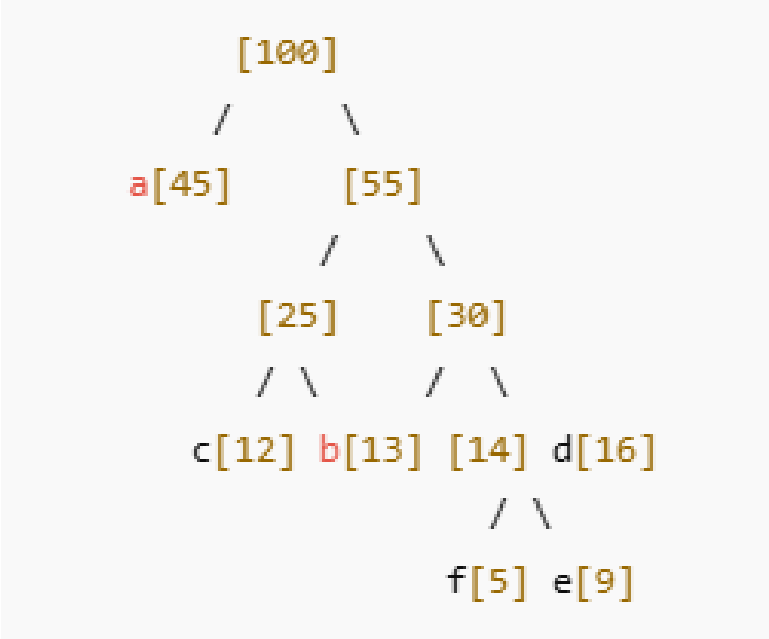
Fixed-length code must have at least 3 bits:

Total number of symbols (total occurrences):45 + 13 + 12 + 16 + 9 + 5 = 100

Each symbol uses 3 bits:100 × 3 = 300 bits

Fixed length storage = 300 bits

Huffman Tree



From this tree we extract the codes (left=0, right=1):

character	a	b	c	d	e	f
frequency	45	13	12	16	9	5
fixed code	000	001	010	011	100	101
Variable code	0	101	100	111	1101	1100

We recalculated each code (frequency × code length) as in the previous example:

a: 45 × 1 = 45

b: 13 × 3 = 39

c: 12 × 3 = 36

d: 16 × 3 = 48

e: 9 × 4 = 36

f: 5 × 4 = 20

45 + 39 + 36 + 48 + 36 + 20 = 224 (Variable length storage = 224 bits).

Compression ratio = 300 / 224 = 1.339

Huffman Coding

Huffman coding is a variable-length coding technique in which shorter codewords are assigned to input values with high probabilities, and longer codewords are assigned to those with low probabilities. The encoder takes fixed-length input characters and produces variable-length output bits, achieving efficient compression.

Advantages of Huffman Coding

- **Efficient lossless compression** when symbol frequencies are different.
- **Simple to implement** and meets practical requirements for time and memory in most applications.
- **Provides a representation close to the minimum entropy** when symbols are independent and their probability distribution is known.
- **Useful in real-time systems**: can be combined with other techniques (such as transforms, RLE, or predictive coding) to improve image/video compression.
- **Flexible**: generates variable-length codes that efficiently exploit statistical redundancy

Limitations and Notes

When Not to Use Huffman Coding

- **Not ideal when symbols are time-correlated**:

  If there are strong relationships or predictable patterns between successive symbols, other methods (such as context-based models combined with arithmetic coding) can achieve better compression.

- **Depends on knowing or estimating symbol frequencies**:

  If the probability distribution of symbols changes frequently, you may need to rebuild the code tree or use dynamic Huffman coding, which increases complexity.

- **Minimum code length is 1 bit**:

  In cases of extremely skewed probabilities, arithmetic coding performs better because it can represent fractional probabilities and achieve compression closer to the theoretical limit.

Example 3:

A small example will make this clear. Suppose we have a 2-bit greyscale image with only four grey levels: 0, 1, 2, 3, with the probabilities 0.2, 0.4, 0.3 and 0.1 respectively. The following table shows fixed length and variable length codes for this image:

Grey value	Probability	Fixed code	Variable code
0	0.2	00	000
1	0.4	01	1
2	0.3	10	01
3	0.1	11	001

Now consider how this image has been compressed. Each grey value has its own unique identifying code. The average number of bits per pixel can be easily calculated as the expected value (in a probabilistic sense):

$$L_{av} = (0.2 * 3) + (0.4 * 1) + (0.3 * 2) + (0.1 * 3) = 1.9$$

Notice that the longest codewords are associated with the lowest probabilities. This average is indeed smaller than 2. This can be made more precise by the notion of entropy, which is a measure of the amount of information

GOOD LUCK EVERYONE