UNIT–5 : Image Compression and Recognition (PYQ Solution)

AKTU · Image Processing (BCS057) · Solutions by **Chirag Jain**, Ycotes

SECTION A Answers (≈70 Words Each)

Q1. Define the use of boundary extraction. (2023)

Boundary extraction is a morphological operation that finds the outline of an object in a binary image: β(A) = A − (A ⊖ B), the set minus its erosion by a structuring element B (usually 3 × 3). It is used to obtain shape descriptors, perimeter and contours for object representation, recognition and measurement.

Q2. What is the need for image compression? (2019)

Images need a large amount of storage and bandwidth; a 1024 × 1024 colour image takes 3 MB. Compression reduces the number of bits needed by removing redundancy (coding, interpixel and psychovisual). It saves storage, speeds up transmission over networks and lowers cost for applications like web, video and medical imaging.

SECTION B Answers (≈250 Words Each)

Q1. Discuss the need for data compression. Explain the run length encoding algorithm in detail. (2023)

Need for compression: raw images and video are very large (a 1-minute HD video is several GB). Compression is needed to **save storage**, **reduce transmission time and bandwidth**, and lower cost. It works by removing **coding redundancy**, **interpixel redundancy** and **psychovisual redundancy**. Compression ratio C = n1 / n2, where n1 and n2 are the bits of the original and compressed data.

Run Length Encoding (RLE) is a lossless method that exploits interpixel redundancy: a sequence of identical consecutive values (a **run**) is replaced by a pair (value, count).

Algorithm:

1. Scan the image row by row (or in a zigzag).
2. Take the first pixel value v and set count = 1.
3. For each next pixel: if equal to v, increment count; else output (v, count), set v to the new value and count = 1.
4. At the end of the row output the last pair.
5. For decoding, repeat each value count times.

Example: the row 5 5 5 5 0 0 0 7 7 9 encodes as (5, 4), (0, 3), (7, 2), (9, 1): 10 values become 8 numbers. For binary images with long white runs, the line 0000000011110000 becomes (0, 8) (1, 4) (0, 4), giving a large saving.



Run length encoding

Advantages: simple, fast, lossless; good for binary images, line drawings, fax.
Disadvantages: poor on noisy or photographic images, since runs are short and the output can be **larger** than the input.

Conclusion: RLE gives high compression only when images contain long runs of identical pixels.

Q2. Describe the Huffman coding algorithm for image compression with an example. (2019)

Huffman coding is a lossless, variable-length coding technique that removes **coding redundancy**: frequent symbols get short codes and rare symbols get long ones. It yields the **smallest possible average code length** among prefix codes for a given symbol set; no code is a prefix of another.

Algorithm:

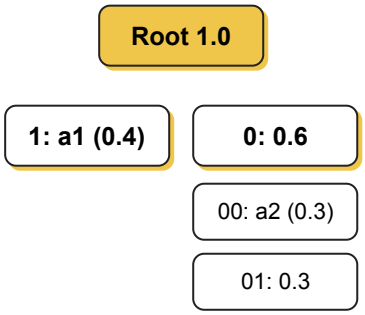
1. Find the probability of each gray level (from the histogram).
2. Sort the symbols in decreasing probability.
3. Combine the two lowest-probability symbols into one, adding their probabilities; re-sort. Repeat until one symbol (probability 1) remains.
4. Assign 0 and 1 to the two branches at every merge (source reduction, then code assignment backwards).
5. The code of a symbol is the bit sequence along the path from the root.

Example: symbols with probabilities a1 = 0.4, a2 = 0.3, a3 = 0.1, a4 = 0.1, a5 = 0.06, a6 = 0.04.

- Merge a5 + a6 = 0.10. List: 0.4, 0.3, 0.1, 0.1, 0.1.
- Merge a3 + a4 = 0.2. List: 0.4, 0.3, 0.2, 0.1.
- Merge 0.2 + 0.1 (a5 + a6 group) = 0.3. List: 0.4, 0.3, 0.3.
- Merge 0.3 + 0.3 = 0.6. List: 0.6, 0.4. Final merge = 1.0.

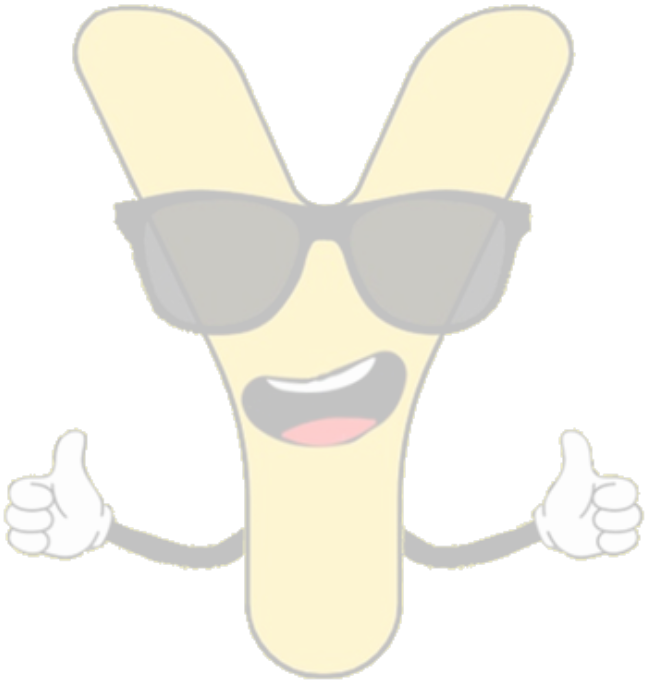
Resulting code lengths: a1 = 1 bit (code 1), a2 = 2 bits (00), a3 = 4 bits (0100), a4 = 4 bits (0101), a5 = 4 bits (0110), a6 = 4 bits (0111).

Average length: L = 0.4(1) + 0.3(2) + 0.1(4) + 0.1(4) + 0.06(4) + 0.04(4) = 0.4 + 0.6 + 0.4 + 0.4 + 0.24 + 0.16 = **2.2 bits/symbol**, against 3 bits for fixed-length coding. Compression ratio = 3 / 2.2 = **1.36**.



Huffman tree (top levels)

Features: instantaneously decodable; the code table must be stored; it is used inside JPEG. **Conclusion:** Huffman coding shortens the average code length to near the entropy of the source.



SECTION C Answers (≈500 Words Each)

Q1. Differentiate between JPEG and MPEG standards. Discuss the JPEG compression process and its steps. (2023, 2019)

JPEG (Joint Photographic Experts Group) is a standard for compressing **still images**. **MPEG (Moving Picture Experts Group)** is a family of standards for compressing **video and audio**.

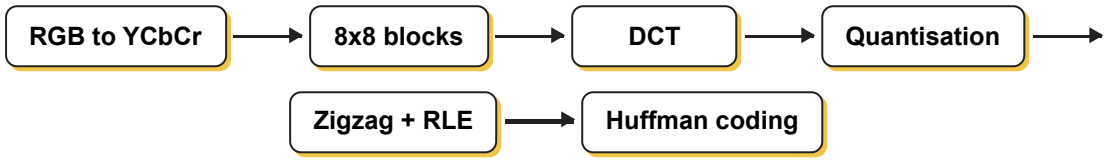
Basis	JPEG	MPEG
Media	Still images	Moving video plus audio
Redundancy used	Spatial (within frame)	Spatial and **temporal** (between frames)
Core method	DCT, quantisation, Huffman	DCT plus **motion estimation and compensation**
Frame types	Single frame	I, P and B frames
Compression ratio	About 10:1 to 20:1	Much higher (50:1 or more)
Complexity	Lower	Higher; needs synchronisation
Versions	Baseline, progressive, JPEG 2000	MPEG-1, -2, -4, H.264
Use	Photographs, web	DVD, TV, streaming

MPEG frames: I-frames are coded independently like JPEG; **P-frames** predict from previous frames; **B-frames** use both past and future frames.

JPEG compression steps (baseline, lossy):

1. **Colour conversion and sub-sampling:** RGB to YCbCr; the chrominance components Cb and Cr are down-sampled (4:2:0) because the eye is less sensitive to colour.
2. **Block splitting:** each component is divided into **8 × 8 blocks**, and 128 is subtracted (level shift).
3. **Forward DCT:** each block is transformed to 64 coefficients: one **DC** coefficient (the average) and 63 **AC** coefficients; energy concentrates in the low frequencies.

4. **Quantisation:** each coefficient is divided by the corresponding entry of a **quantisation table** and rounded. High frequencies get large divisors, so most become zero. This is the **only lossy step**; the quality factor controls the table.
5. **Zigzag scanning:** the 8 × 8 block is read in zigzag order, producing long runs of zeros at the end.
6. **Entropy coding:** DC coefficients are coded with **differential coding (DPCM)** against the previous block; AC coefficients with **run-length coding** (zero run, value) and then **Huffman coding**.
7. The compressed bit stream is written with header and tables.



JPEG encoder

Decoding reverses the steps: Huffman decode, de-zigzag, de-quantise, inverse DCT, up-sample and colour-convert.

Conclusion: JPEG compresses a single image using DCT, while MPEG adds motion compensation to exploit similarity between frames.

Q2. Differentiate between image compression and image recognition briefly. (2023)

Image compression reduces the amount of data required to represent an image, whereas **image recognition** identifies the objects or patterns in an image and assigns them labels.

Basis	Image compression	Image recognition
Purpose	Reduce storage and transmission size	Identify and classify objects
Input	Whole image	Segmented object or features
Output	Compact bit stream (image back after decoding)	Class label or description
Level	Low-level processing	High-level processing
Techniques	Huffman, RLE, DCT, JPEG	Feature extraction, classifiers (minimum distance, neural network), pattern matching
Measure	Compression ratio, PSNR	Recognition accuracy
Information	Preserves image content	Extracts meaning, discards pixel detail
Examples	JPEG, PNG	Face, fingerprint, character recognition

Compression may be lossless (exact reconstruction) or lossy. **Recognition** involves **patterns** (arrangements of descriptors), **feature vectors** and decision functions that assign an unknown pattern to one of several classes.



Compression versus recognition

Conclusion: compression is about efficient representation, and recognition is about understanding the content.
