UNIT–2 : Image Enhancement (PYQ Solution)

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

SECTION A Answers (≈70 Words Each)

Q1. Differentiate between image enhancement and image restoration. (2023, 2019)

Enhancement improves the visual appearance of an image for a particular purpose; it is subjective and needs no knowledge of degradation (contrast stretching, sharpening). Restoration attempts to recover the original image by reversing a known degradation (blur, noise) using a mathematical model; it is objective (inverse filter, Wiener filter). Enhancement may add distortion; restoration tries to remove it.

Q2. What is the importance of histogram equalization? (2019)

Histogram equalisation spreads the gray levels using the transformation s = T(r) = (L − 1) Σ pr(rj), the cumulative distribution function. It produces an image with an approximately uniform histogram, which increases global contrast and reveals detail in dark or washed-out images. It is automatic, needs no parameters and is useful in medical and satellite imaging.

Q3. What is the role of homomorphic filtering? (2019)

Homomorphic filtering improves an image affected by non-uniform illumination. Since f = i · r, taking the log makes ln f = ln i + ln r. A high-frequency-emphasis filter in the frequency domain attenuates low-frequency illumination and boosts high-frequency reflectance; then exp restores the image. The result is compressed dynamic range and enhanced contrast.

SECTION B Answers (≈250 Words Each)

Q1. Explain piecewise linear transformations of image enhancement with suitable examples. (2023)

Piecewise linear transformations use a function made of linear segments, s = T(r), where r is the input and s the output gray level. Their advantage is that the form can be arbitrarily complex and controlled by the user; the disadvantage is more user input.

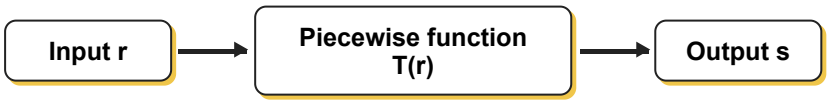
1. Contrast stretching: low-contrast images (poor illumination or limited sensor range) are expanded to use the full gray range using control points (r1, s1) and (r2, s2).

- If r1 = r2 the transform is thresholding.
- If (r1, s1) = (rmin, 0) and (r2, s2) = (rmax, L − 1) the image is stretched linearly.
- Example: r in [50, 150] stretched to [0, 255] gives s = (r − 50) × 255 / 100; r = 100 maps to 127.5, about 128.

2. Gray-level (intensity) slicing: highlights a specific range [A, B] of gray levels.

- Without background: s = L − 1 for A ≤ r ≤ B, else 0 (binary output).
- With background: s = L − 1 inside the range, other values preserved.
- Application: highlighting masses in X-ray images or water bodies in satellite pictures.

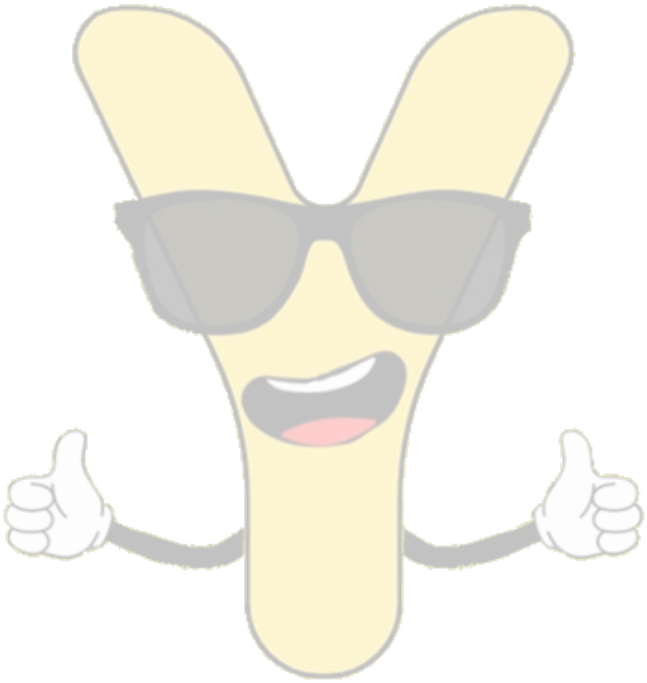
3. Bit-plane slicing: an 8-bit image is a stack of eight 1-bit planes. Plane 7 (MSB) carries most visual information; low planes carry fine detail and noise. Useful in compression and in finding the number of bits needed.



Piecewise linear mapping

Transformation	Purpose
Contrast stretching	Increase dynamic range
Intensity slicing	Highlight a range
Bit-plane slicing	Analyse bit importance

Conclusion: piecewise linear functions give flexible, user-controlled enhancement.



SECTION C Answers (≈500 Words Each)

Q1. Differentiate correlation and convolution with a 1-D function and filter example. (2023)

Both slide a filter (mask) w over an image or signal f and compute a sum of products at each position.

- **Correlation:** $g(x) = \Sigma\, w(s)\, f(x + s)$. The mask is used as it is.
- **Convolution:** $g(x) = \Sigma\, w(s)\, f(x - s)$. The mask is first **rotated (flipped) by 180°**.

Example: let f = [0 0 0 1 0 0 0 0] (a unit impulse at index 3) and the filter w = [1 2 3] (centre = 2, offsets −1, 0, +1).

- **Correlation:** $g(x) = 1 \cdot f(x-1) + 2 \cdot f(x) + 3 \cdot f(x+1)$. At x = 2: 3; at x = 3: 2; at x = 4: 1. Result: **[0 0 3 2 1 0 0 0]**, a rotated copy of w.
- **Convolution:** flip w to [3 2 1], so $g(x) = 3 \cdot f(x-1) + 2 \cdot f(x) + 1 \cdot f(x+1)$. At x = 2: 1; at x = 3: 2; at x = 4: 3.
  Result: **[0 0 1 2 3 0 0 0]**, an exact copy of w (the impulse response).

Basis	Correlation	Convolution
Mask flipping	No	Yes (180° rotation)
Output for an impulse	Rotated mask	Same mask
Commutative	No	Yes
Associative	No	Yes
Use	Template matching	Filtering, linear systems theory

For a **symmetric** mask (e.g. averaging, Gaussian) both give identical results. Convolution in the spatial domain equals multiplication in the frequency domain, $f * w \Leftrightarrow F \cdot W$, which is why it is central to filtering.

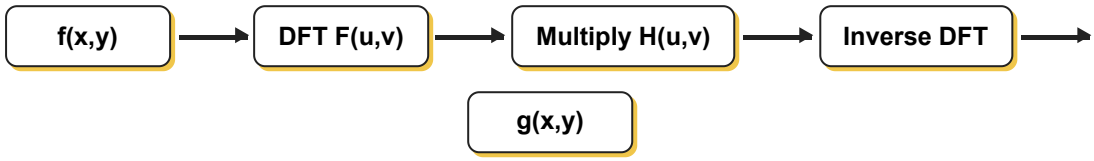
Conclusion: the only difference is the flip of the mask.

Q2. Explain the process of filtering in the frequency domain. Discuss low-pass and high-pass filters. (2023)

Frequency-domain filtering modifies the Fourier transform of the image.

Steps:

1. Multiply the input f(x, y) by (−1)^(x+y) to centre the spectrum.
2. Compute the 2-D DFT, F(u, v).
3. Multiply by a filter transfer function H(u, v): G(u, v) = H(u, v) F(u, v).
4. Compute the inverse DFT of G.
5. Take the real part and multiply again by (−1)^(x+y).



Frequency-domain filtering

Low-pass filters (smoothing): pass low frequencies, attenuate high frequencies (edges, noise), so the image is blurred. D0 is the cut-off distance from the origin and D(u, v) the distance from the centre.

- **Ideal LPF:** H = 1 if D(u, v) ≤ D0, else 0. Sharp cutoff causes **ringing**.
- **Butterworth LPF:** H = 1 / [1 + (D/D0)^(2n)]; smooth transition, little ringing for low order n.
- **Gaussian LPF:** H = exp(−D²/2D0²); no ringing.

High-pass filters (sharpening): H_hp = 1 − H_lp. They pass high frequencies and remove low ones, enhancing edges and fine detail.

- **Ideal HPF:** H = 0 if D ≤ D0, else 1 (ringing).
- **Butterworth HPF:** H = 1 / [1 + (D0/D)^(2n)].
- **Gaussian HPF:** H = 1 − exp(−D²/2D0²).

Basis	Low-pass	High-pass
Passes	Low frequencies	High frequencies
Effect	Blur, noise reduction	Edge sharpening
Relation	H_lp	1 − H_lp

Conclusion: frequency-domain filters give precise control over which image components are kept.

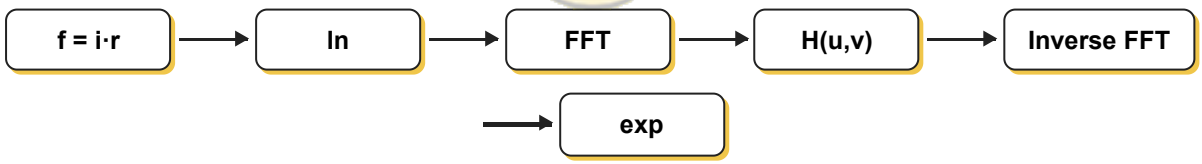
Q3. Write notes on bit-plane slicing and the homomorphic filter. (2023)

(i) Bit-plane slicing: an image with 8 bits per pixel can be seen as eight 1-bit planes, from plane 0 (least significant bit) to plane 7 (most significant bit).

- Plane 7 (MSB) holds most of the visually significant data and looks like a thresholded copy of the image.
- Lower planes hold fine details and noise.
- Plane k of pixel value r: bit k of its binary representation, e.g. r = 150 = 10010110, so plane 7 = 1, plane 4 = 1, plane 0 = 0.
- **Uses:** finding which bits matter, **compression** (store only upper planes), steganography, image analysis.
- **Reconstruction:** f = Σ 2^k × plane k; using the top 4 planes gives a good approximation with half the storage.

(ii) Homomorphic filter: improves an image with uneven illumination by treating it as f(x, y) = i(x, y) · r(x, y), where i = illumination (slowly varying, low frequency) and r = reflectance (detail, high frequency). Products cannot be separated by a linear filter, so:

1. Take the logarithm: ln f = ln i + ln r.
2. Take the Fourier transform: Z(u, v) = F_i(u, v) + F_r(u, v).
3. Apply a filter H(u, v) with **γL < 1 at low frequencies** (reducing illumination) and **γH > 1 at high frequencies** (boosting reflectance): H = (γH − γL)[1 − exp(−c D²/D0²)] + γL.
4. Take the inverse Fourier transform.
5. Take the exponential to return to the image domain.



Homomorphic filtering

Result: dynamic range compression and contrast enhancement together; used for photographs with shadows and X-ray images.

Q4. What is spatial filtering? Explain the difference between smoothing

and sharpening spatial filters with examples. (2019)

Spatial filtering modifies an image by operating directly on pixels in a neighbourhood. A small mask (kernel, e.g. 3 × 3) is moved over the image, and at each position the response is the sum of products of mask coefficients and the pixels beneath: $g(x, y) = \Sigma \Sigma w(s, t) f(x + s, y + t)$. Linear filters use sums of products; non-linear filters (median, max) use ranking.

Smoothing filters (low-pass) reduce noise and blur small detail; each output is an average of neighbours.

- **Box (averaging) filter:** all coefficients 1/9.
- **Weighted average:** (1/16) × [1 2 1; 2 4 2; 1 2 1], giving more weight to the centre.
- **Median filter (non-linear):** replaces the pixel with the median; excellent for salt-and-pepper noise.

Sharpening filters (high-pass) highlight fine detail and edges, based on **derivatives**.

- **Laplacian** (second derivative): mask [0 1 0; 1 −4 1; 0 1 0]; sharpened image $g = f - \nabla^2 f$ (for negative-centre mask).
- **Gradient (Sobel):** $|\nabla f| \approx |Gx| + |Gy|$, mask Gx = [−1 0 1; −2 0 2; −1 0 1].
- **Unsharp masking / high-boost:** g = f + k (f − f_blurred).

Example: on the row 10 10 10 80 10 10, a 3-point average replaces 80 by (10 + 80 + 10)/3 = 33 (noise reduced, but edges blur), while a Laplacian response at the same point is 10 − 160 + 10 = −140, strongly marking the spike.

Basis	Smoothing	Sharpening
Operation	Averaging, integration	Differentiation
Coefficients	Positive, sum to 1	Mixed sign, sum to 0
Effect	Blurs, removes noise	Enhances edges
Examples	Box, Gaussian, median	Laplacian, Sobel, unsharp

Conclusion: smoothing removes noise at the cost of detail; sharpening restores detail but amplifies noise.
