MSE Question Bank

Chapter 1 :

1. Pixel Representation and Resolution

a. Define a pixel and explain its role in digital image representation.

A pixel, short for "picture element," is the smallest unit of a digital image. It represents a single point in the image and is typically a tiny square or dot that contains color and intensity information. Pixels collectively form a grid or array that constructs the entire image when viewed together. In digital image representation, each pixel is assigned numerical values (e.g., RGB values for color images or grayscale intensity for monochrome images), which define its appearance. The role of pixels is fundamental—they serve as the building blocks that encode visual information, allowing images to be stored, processed, and displayed on digital devices. The finer the pixel grid (i.e., more pixels), the more detailed the image appears, though this increases storage and processing demands.

b. What is image resolution? Discuss the impact of resolution on the quality of an image.

Image resolution refers to the level of detail an image holds, typically measured as the number of pixels in width × height (e.g., 1920×1080) or as pixels per inch (PPI/DPI). Higher resolution means more pixels are used to represent the image, resulting in sharper, more detailed visuals. For example, a 4K image (3840×2160) has four times the pixels of a 1080p image (1920×1080), offering greater clarity. Low resolution, conversely, leads to pixelation—visible blockiness—because fewer pixels approximate the image's features. Resolution impacts quality by determining how accurately fine details, edges, and textures are rendered. However, higher resolution increases file size and computational requirements, making it a trade-off between quality and practicality in applications like web design or printing.

---

2. Color and Intensity

a. What is intensity in an image? How is it related to the pixel values?

Intensity in an image refers to the brightness or lightness of a pixel, often represented as a single value in grayscale images or as a combination of values in color images. In a grayscale image, intensity ranges from 0 (black) to 255 (white) in an 8-bit system, directly corresponding to the pixel's numerical value. In color images (e.g., RGB), intensity is derived from the combined effect of red, green, and blue channel values, though it's often separated in models like HSI (Hue, Saturation, Intensity). Pixel values dictate intensity—higher values mean brighter pixels, while lower values mean darker ones. Intensity is key to contrast and visibility, influencing how details stand out in an image.

b. Explain the concepts of Hue, Saturation, and Brightness in terms of color representation in images.

Hue, Saturation, and Brightness (HSB, also called HSV—Value) are components of a color model used to represent colors in images more intuitively than RGB. Hue refers to the color type (e.g., red, blue, green), measured as an angle on a color wheel (0°–360°). Saturation describes the purity or vividness of the color—100% saturation is a fully vibrant color, while 0% is gray, regardless of hue. Brightness (or Value) indicates the lightness or darkness of the color, ranging from 0% (black) to 100% (fully bright). In image processing, HSB separates color information (hue) from intensity (brightness) and richness (saturation), making it useful for tasks like color adjustment or segmentation, where isolating specific attributes enhances control over the image's appearance.

---

3. Image Formats and Storage

a. Discuss the commonly used image formats (JPEG, PNG, TIFF, BMP) and their characteristics.

JPEG (Joint Photographic Experts Group) is a lossy compression format ideal for photographs, reducing file size by discarding minor details, but it sacrifices some quality. PNG (Portable Network Graphics) uses lossless compression, preserving all data, and supports transparency, making it great for web graphics with sharp edges. TIFF (Tagged Image File Format) is a flexible, lossless format often used in professional settings like printing or archiving, supporting high quality and metadata but resulting in large files. BMP (Bitmap) is an uncompressed format storing raw pixel data, offering maximum quality but impractical for storage due to huge file sizes. Each format balances quality, size, and use case—JPEG for photos, PNG for graphics, TIFF for editing, BMP for simplicity.

b. Explain the importance of image storage in terms of size and quality.

Image storage is critical because it determines how efficiently data is saved and accessed while maintaining visual fidelity. File size depends on resolution, color depth, and compression—high-resolution, uncompressed images (e.g., BMP) demand more storage but retain perfect quality, while compressed formats (e.g., JPEG) save space at the cost of detail loss. Quality matters for intended use: professional editing needs high-quality, lossless storage (e.g., TIFF), while web images prioritize small sizes (e.g., JPEG). Efficient storage optimizes disk space and transmission speed, but over-compression can introduce artifacts like blurriness or blockiness, degrading the image. Balancing size and quality ensures usability without compromising essential visual information.

---

4. Connectivity and Region Representation

a. Define connectivity in an image. How is pixel connectivity useful in image processing?

Connectivity in an image describes how pixels are related or grouped based on their adjacency. Common types are 4-connectivity (pixels sharing an edge—up, down, left, right) and 8-connectivity (including diagonals). It determines whether pixels belong to the same object or region. In image processing, connectivity is vital for tasks like object detection, boundary tracing, and segmentation. For example, identifying a shape involves grouping connected pixels with similar properties (e.g., intensity). Algorithms like connected-component labeling use connectivity to isolate distinct regions, enabling analysis of shapes, patterns, or anomalies in applications like medical imaging or machine vision.

b. Explain the concept of region representation in images and its importance in segmentation tasks.

Region representation involves defining and describing areas of an image with uniform properties, such as intensity, color, or texture. It can be boundary-based (outlining edges) or region-based (grouping interior pixels). In segmentation, regions are isolated to separate objects from backgrounds or distinguish features—like tumors in MRI scans. Representation methods include chain codes for boundaries or adjacency matrices for pixel groups. Its importance lies in simplifying complex images into meaningful parts, enabling object recognition, measurement, or classification. Accurate region representation enhances segmentation precision, critical for tasks like autonomous driving or facial recognition, where distinct areas must be reliably identified.

---

5. Mathematical and Logical Operations

a. What are mathematical operations in image processing? Provide examples.

Mathematical operations in image processing manipulate pixel values using arithmetic or statistical methods to alter or analyze images. Examples include addition (brightening an image by adding a constant to pixel values), subtraction (detecting differences between two images, e.g., motion detection), multiplication (scaling intensity), and convolution (applying filters like blurring or edge detection). For instance, adding 50 to all pixel values in a grayscale image increases brightness, while convolving with a Sobel kernel highlights edges. These operations transform images for enhancement, feature extraction, or noise reduction, forming the backbone of many processing techniques.

b. Explain logical operations in image processing with relevant examples. How do these differ from mathematical ones?

Logical operations use Boolean logic (AND, OR, NOT, XOR) on pixel values, typically in binary images, to combine or modify regions. For example, ANDing two binary images (1 for object, 0 for background) isolates overlapping areas, useful in masking. OR combines regions from two images, while XOR highlights differences (e.g., detecting changes between frames). Unlike mathematical operations, which adjust numerical values continuously (e.g., adding 10 to intensity), logical operations work on discrete states (true/false, 1/0), focusing on spatial relationships rather than intensity manipulation. They're essential for tasks like shape analysis or region selection, differing in their binary, condition-based approach.

---

6. Histogram Processing and Image Quality

a. What is histogram processing? How is it used for image enhancement?

Histogram processing involves analyzing and modifying the distribution of pixel intensity values in an image, represented as a histogram (x-axis: intensity, y-axis: frequency). For enhancement, techniques like histogram equalization redistribute intensities to increase contrast—stretching the range so dark and bright areas are more distinct. For example, a low-contrast image with a narrow histogram becomes vivid after equalization spreads values across 0–255. It's used to improve visibility in underexposed photos or medical images, revealing hidden details without altering spatial content, making it a powerful, simple enhancement tool.

b. Discuss the factors affecting image quality. How do image noise and aliasing affect image quality?

Image quality depends on resolution, color depth, compression, noise, and aliasing. Noise—random pixel variations (e.g., graininess from low light)—reduces clarity, obscuring details and introducing artifacts like speckles. Aliasing occurs during sampling when high-frequency details (e.g., fine edges) are undersampled, causing jagged "staircase" effects or moiré patterns. Both degrade quality: noise adds unwanted texture, while aliasing distorts shapes. High resolution and proper sampling mitigate aliasing, while noise reduction filters (e.g., Gaussian smoothing) improve smoothness, though over-filtering can blur details. Quality balances fidelity to the original scene with processing artifacts.

---

7. Image Sampling and Quantization

a. What is image sampling? Discuss its importance in converting an analog image to a digital one.

Image sampling is the process of converting a continuous analog image (e.g., a photograph) into a discrete grid of pixels by measuring intensity at regular intervals. It determines the spatial resolution—more samples (higher pixel count) capture finer details, while fewer samples risk losing information. Sampling is crucial because it discretizes the infinite analog signal into a manageable digital format for storage and processing. The Nyquist-Shannon theorem states sampling must be at least twice the highest frequency to avoid aliasing. Proper sampling ensures the digital image faithfully represents the analog original, critical for applications like digital photography or video.

b. Define quantization and explain how it affects the representation of an image.

Quantization is the process of mapping continuous intensity values from sampling into a finite set of discrete levels (e.g., 0–255 in 8-bit images). It reduces the infinite range of analog intensities to a manageable number, defining color depth or grayscale levels. Higher quantization levels (e.g., 16-bit, 65,536 levels) preserve subtle variations, enhancing quality, while low levels (e.g., 2-bit, 4 levels) cause posterization—banding or loss of smooth gradients. Quantization affects file size and visual fidelity: too few levels degrade realism, while too many increase storage needs. It's a trade-off between precision and practicality in digital representation.

---

8. Geometric Transformations

a. Describe the basic geometric transformations used in image processing (translation, rotation, scaling).

Geometric transformations alter an image's spatial layout. Translation shifts the image by a fixed distance (e.g., moving it 50 pixels right), defined by adding offsets to pixel coordinates. Rotation pivots the image around a point (e.g., 90° clockwise), using trigonometric functions (sine, cosine) to compute new positions. Scaling resizes the image—enlarging (multiplying coordinates by >1) or shrinking (<1)—adjusting its dimensions. These operations use matrices for efficiency (e.g., a 2×3 matrix for affine transforms). They're essential for aligning images, correcting distortions, or resizing content in editing software, preserving pixel values while changing their arrangement.

b. How does the aspect ratio of an image influence its visual representation during transformations?

Aspect ratio (width-to-height ratio, e.g., 4:3) defines an image's proportional shape. During transformations like scaling, maintaining the aspect ratio ensures the image doesn't distort—uneven scaling (e.g., stretching width more than height) warps objects, making circles ovals or faces unnatural. For example, resizing a 4:3 image to 16:9 without cropping or padding squashes or stretches content. Rotation and translation don't affect aspect ratio directly, but resizing within a fixed frame may require adjustments (e.g., letterboxing). Preserving aspect ratio maintains visual integrity, critical for realistic representation in displays or design.

---

10. Image Smoothing and Sharpening using Frequency Domain Filters

a. What are Ideal, Butterworth, and Gaussian filters in the frequency domain? Explain their role in image smoothing and sharpening.

Ideal filters sharply cut off frequencies—low-pass (smoothing) keeps low frequencies, high-pass (sharpening) keeps high ones—but cause ringing artifacts due to abrupt transitions. Butterworth filters offer a smoother cutoff, balancing attenuation and smoothness; low-pass reduces noise, high-pass enhances edges with fewer artifacts than Ideal filters. Gaussian filters use a bell-shaped curve, gradually tapering frequencies; low-pass blurs for smoothing, high-pass boosts details for sharpening, avoiding ringing entirely. In the frequency domain, these filters modify the DFT spectrum, controlling which features (smooth areas or edges) dominate, tailoring the image's appearance.

b. Discuss the effect of these filters on an image and their use in noise reduction and enhancement.

Low-pass filters (Ideal, Butterworth, Gaussian) smooth images by removing high-frequency noise (e.g., speckles), but Ideal's sharp cutoff may blur edges excessively, while Gaussian preserves them better. High-pass filters sharpen by amplifying edges, enhancing contrast, though Ideal filters may amplify noise too. Butterworth strikes a middle ground, reducing noise while retaining some detail. In practice, Gaussian low-pass filters denoise photos, while high-pass versions enhance text or features in medical images. The choice depends on the noise level and desired clarity—smoothing trades detail for cleanliness, sharpening boosts visibility but risks noise exaggeration.

---

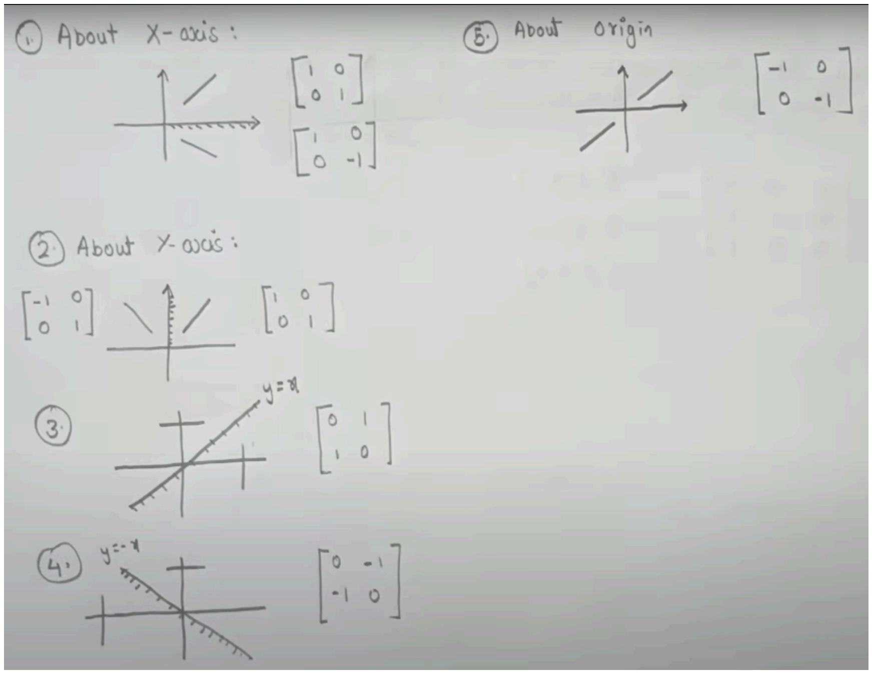
14. Analyse how distance measures can be used in tasks like segmentation or pattern recognition.

Distance measures play a critical role in segmentation and pattern recognition by quantifying how similar or dissimilar data points are, enabling effective grouping or classification. In segmentation, the goal is to partition data—such as pixels in an image or points in a dataset—into meaningful clusters or regions. Euclidean distance, the straight-line metric, is widely used in tasks like k-means clustering for image segmentation, grouping pixels by color or position, though it assumes spherical clusters and is sensitive to scale. Manhattan distance, summing absolute differences, fits grid-like data (e.g., urban layouts) and resists outliers better. Mahalanobis distance, incorporating covariance, excels in separating correlated or variably shaped regions, such as tissues in medical imaging.

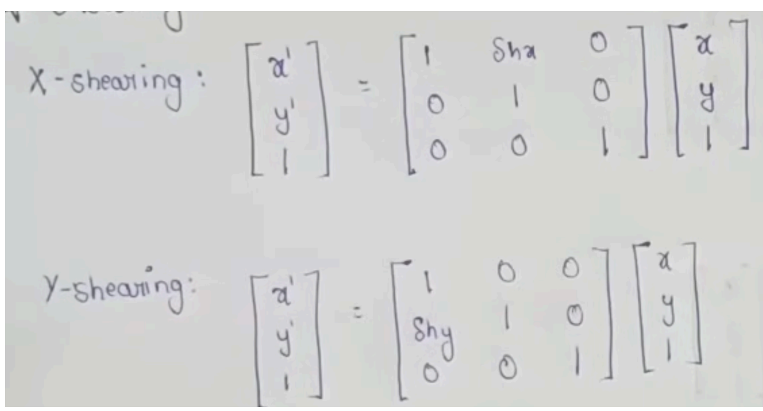
In pattern recognition, distance measures help classify or identify patterns by comparing an input to known examples. Cosine similarity, focusing on vector angles, is ideal for high-dimensional data like text or face recognition, emphasizing direction over magnitude. Hamming distance counts mismatches in binary data, perfect for tasks like error detection or feature matching in images (e.g., SIFT descriptors). For sequential data, Dynamic Time Warping (DTW) aligns time-series—think speech or gesture recognition—adapting to timing variations. Each metric suits specific data: Euclidean for continuous, isotropic spaces; cosine for sparse, directional patterns.

Chapter 2 :

1. Translate polygon with A(2,7) , B(7,10) and C (10,2) by 3 unit in X direction and 4 unit in Y direction.
2. Scale a polygon with co-ordinates A(2,5), B(7,10) and C(10,2) by 2 units in X-direction and 3 units in Y direction.
3. Determine the transformation matrix of triangle A(4,1), B(5,2) and C(4,3) about the line x=0 and determine the resultant co-ordinates.



4. Shear a polygon A(0,0), B(1,0), C(1,1) and D(0,1) by shearing vector dhx=2 and determine its new co-ordinates.



5. A point (4,3) is rotated counter clockwise by an angle 45. Find the rotation matrix and the resultant point.
6. Translate a square ABCD A(0,0),B(3,0), C(3,3) and D(0,3) by 2 units in both direction then scale it by 1:5 units in X-direction and 0.5 in Y direction. Determine the resultant co-ordinated of polygon.

Chapter 3 :

1. What is the difference between spatial domain and frequency domain representations of an image?

Aspect	Spatial Domain	Frequency Domain
Definition	Data by position (x, y) or time (t).	Data by frequency components.
Form	Amplitude at coordinates.	Magnitude/phase of frequencies.
Basis	Raw values, e.g., f(x, y).	Fourier Transform, e.g., F(u, v).
Operations	Direct edits (e.g., pixel changes).	Frequency filtering (e.g., multiplication).
Scope	Local (specific points).	Global (whole signal/image).
Tasks	Brightness, cropping.	Noise removal, compression.
Example	Set pixel (100, 150) to 200.	Cut frequencies > 10 cycles/pixel.
Tools	Arithmetic, convolution kernels.	FFT, filters.
Pros	Simple, intuitive.	Efficient for patterns, filtering.
Cons	Limited for frequency tasks.	Needs transform, less direct.

---

2. Explain the significance of the Discrete Fourier Transform (DFT) in image processing.

The DFT converts an image from the spatial domain to the frequency domain, decomposing it into a sum of sinusoidal components (frequencies). Its significance includes:

- **Filtering**: Separates low (smooth areas) and high (edges) frequencies for smoothing or sharpening.
- **Analysis**: Reveals periodic patterns or noise characteristics.
- **Compression**: Identifies dominant frequencies, aiding data reduction.

- **Applications**: Noise removal, edge detection, and image restoration rely on DFT's ability to isolate and manipulate frequency components.

---

3. How does the Discrete Cosine Transform (DCT) differ from the Discrete Fourier Transform (DFT)?

- **Basis Functions**: DFT uses complex exponentials (sines and cosines with phase), while DCT uses only cosine functions, producing real-valued outputs.
- **Symmetry**: DCT assumes the input is symmetrically extended, doubling the effective length and concentrating energy in fewer coefficients; DFT does not.
- **Output**: DFT yields complex numbers (magnitude and phase); DCT yields real numbers, simplifying computations.
- **Energy Compaction**: DCT compacts energy better than DFT, making it more efficient for compression (e.g., JPEG).
- **Boundary Effects**: DCT reduces edge artifacts due to its implicit symmetry assumption.

---

4. What is the purpose of the Discrete Wavelet Transform (DWT) in image processing?

The DWT decomposes an image into wavelet coefficients, capturing both frequency and spatial location information. Its purposes include:

- **Compression**: Provides multi-resolution analysis, concentrating energy in fewer coefficients (e.g., JPEG2000).
- **Denoising**: Separates noise (high-frequency) from signal at different scales, enabling precise removal.
- **Feature Extraction**: Identifies localized features (e.g., edges) across scales, useful in pattern recognition.
- **Advantage**: Unlike DFT's global analysis, DWT's localized nature suits non-stationary signals like images.

---

5. Derive the formula for the 2D Discrete Fourier Transform (DFT) of an image.

https://grok.com/share/bGVnYWN5_f34d15e8-f531-47e5-8326-c2d56d87a849

6. Given an image, explain how to compute its DFT.

https://grok.com/share/bGVnYWN5_6113e71d-8ba9-4eb9-a134-13f7245ac8ef

7. Write down the mathematical representation of the 1D Discrete Cosine Transform (DCT).

For a 1D signal $f(x)$ of length N, the DCT (Type-II, most common in image processing) is:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

where:

- $u = 0, 1, ..., N-1$ (frequency index),
- $\alpha(u) = \sqrt{\frac{1}{N}}$ for $u = 0$, and $\alpha(u) = \sqrt{\frac{2}{N}}$ for $u > 0$ (normalization factor).
- **Inverse DCT**:

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

In 2D (e.g., JPEG), it's applied separably to rows and columns.

8. What is the frequency response of a Gaussian filter, and how does it affect image processing?

9. How can you use frequency domain filters (such as Ideal, Butterworth, or Gaussian filters) to perform image smoothing or sharpening?

11. What are the benefits of using the Discrete Cosine Transform (DCT) in image compression (e.g., JPEG)?

- **Energy Compaction**: DCT concentrates most signal energy in fewer low-frequency coefficients, allowing efficient truncation of high-frequency terms.
- **Real Output**: Produces real numbers (vs. DFT's complex), simplifying storage and computation.
- **Symmetry**: Reduces boundary artifacts, improving visual quality.
- **Block Processing**: Applied to 8×8 blocks in JPEG, enabling localized compression and parallel processing.
- **Efficiency**: Pairs with quantization and entropy coding to achieve high compression ratios with minimal perceptual loss.

---

12. In what situations would you prefer to use the Discrete Wavelet Transform (DWT) over the DFT for image compression or denoising?

- **Compression**: Prefer DWT (e.g., JPEG2000) for better quality at high compression ratios due to multi-resolution analysis and superior energy compaction, avoiding block artifacts (unlike DCT/DFT).
- **Denoising**: Use DWT when noise varies spatially or across scales (e.g., natural images), as it isolates features locally, unlike DFT's global frequency approach.
- **Non-Stationary Signals**: DWT excels with images having abrupt changes (e.g., edges), while DFT assumes stationarity, making it less effective for such cases.