Q1. Define computer vision system & Explain components of its to parallel human system

Ans 1 - A CV is a technology that enables computer to "see" & interpret visual data from images or videos, much like how human use their eyes and brain to understand world.

Compones of its to parallel human system

1) Camera (Human Eyes) - The camera in a CV system captures images or videos, similar to how human eye capture visual information. Cameras act as the "sensors" to record what's in front of them, like our eyes do.

2) Image processing algorithms (Brain visual cortex):- After capturing images, the system uses image processing algorithms to analyze & understand the content. This is like how the brain processes signals from the eyes. The system might detect edges, recognize objects or classify images much like how brain interprets what we see.

3) Feature extraction (Recognition in Brain):- In this step, specific features are identified from the image such as shapes, colors or patterns. Similarly, when we see something, our brain picks up on key features to recognize objects

4) Decision - making or Output (Human understanding / Actions) Once the computer understand the image, it make decisions, like identifying an object or guiding a robot. Human also act based on visual information, such as recognizing a person & saying hello.

Q2. List different file formats available for the CV system?

Ans 2. Common file formats available for the CV system are :-

1. IMAGE FORMATS :-

-> JPEG (or JPG) :- widely used for storing compressed images; great for reducing file size while maintaining decent quality.

-> PNG :- Supports lossless compression & transparency, used when image quality is crucial.

-> BMP :- Uncompressed format; large file size but retains high image quality.

-> TIFF :- Used for high-quality images, often in medical or scientific fields; supports lossless compression.

-> GIF :- Support animations & simple graphics, limited to 256 colors.

-> RAW :- Unprocessed data from camera sensors, retaining maximum image quality, used in professional photography.

2. VIDEO FORMATS :-

-> MP4 :- Commonly used for storing compressed video with good quality; widely supported across devices.

-> AVI :- An older format that stores video & audio; can be uncompressed or compressed.

-> MKV :- supports multiple audio tracks, subtitles & video streams.

-> MOV :- Develop by Apple, used for high quality videos.

-> FLV :- used for online video streaming; flash-based.

3. 3D Image Formats :-

-> PLY :- store 3D data in the form of points, lines & polygons.

-> OBJ :- Used for 3D models, often in simulations & virtual reality.

-> STL :- Commonly used for 3D printing & modeling.

4. Annotation Formats (for labeling data in CV tasks).

-> XML (PASCAL VOC) :- Used to store data i.e labeled data for object detection & classification task.

→ JSON (COCO) :- stores annotations for large scale datasets, including bounding boxes & key points for tasks like object detection & pose estimation

→ YOLO TXT :- Simple text format used in YOLO object detection frameworks for bounding box annotations.

Q3. What are different energy forms used in CV system & list applications?

Ans- Different energy forms are :-

1. Visible Light :- The most common form of energy. Cameras capture light in visible spectrum, similar to how human eye works.

Applications :-

→ Object detection & recognition :- Identifying objects in images & videos.
→ Facial Recognition :- Used in security system, smartphones & authentication.
→ Autonomus Vehicles :- Detecting pedestrain, signs & obstacles.
→ Medical Imaging :- Used in standard imaging techniques like digital pathology.

2. Infrared (IR) - Infrared light is used for imaging in low-light or dark environments. It captures information based on heart rather than visible light.

Applications :-

→ Night vision : Used in security system, wildlife monitoring, & military surveillance
→ Gesture recognition :- Infrared cameras are used in gaming to detect body movements.
→ Eye - Tracking :- Used in augmented reality (AR) and virtual reality (VR) system to track eye movement.
→ Thermography :- Used in building inspections & medical diagnostic to detect temperature variations.

3. Thermal Radiation : Thermal cameras detect heat emissions from objects, creating a thermal map rather than an image based on light.

Applications :

→ Fire detection :- Used in monitoring systems to detect heat & fire hazards.

→ Search and rescue :- Thermal cameras help locate people in low-visibility conditions (eg. smoke, darkness).

→ Industrial Inspection :- Used for monitoring equipment overheating or detecting electrical faults.

→ Healthcare :- Detecting body temperature variations for fever screening medical diagnostic.

4. Ultra sound : ultra sound waves are used to capture visual information in certain specialized fields. The reflected sound waves are used to create images.

Applications :-

→ Medical Imaging (Ultrasound Scans) : Commonly used to visualize internal organs or monitor fetal development.

→ Obstacle detection in robotics : Used in autonomous robots to detect nearby objects similar to sonar

5. X-rays : X-rays penetrate solid objects to provide images of internal structures

Applications :-

→ Medical Imaging (X-rays, CT scans) :- Used to capture images of bones, tissues, and internal structures in the body.

→ Security Scanning :- Used in airports & customs to scan luggage & detect hidden items

→ Industrial Inspection :- Used for non-destructive testing of materials & products

6. LIDAR (Light detection & Ranging) :- LIDAR uses laser light to measure distances and create 3D maps of environments by analyzing the reflected laser signals.

Applications :

→ Autonomous vehicles : Used to detect objects & create real-time 3D maps of surroundings for navigation.

→ Topography & Mapping : Used to create high precision maps in geographic surveys.

→ Drone-based Inspection : LIDAR sensors are used on drones for mapping large areas, forests & urban environments.

7. Radio waves :- Radio waves can penetrate walls and are used to detect objects or movements in certain envt.

Applications :

→ Radar Systems : Used in vehicles for adaptive cruise control & collision avoidance.

→ Motion detection : Used in security system to detect motion behind walls or in large areas.

Q4. What are the applications of the CV system?

Ans Applications of the CV system

1. Heathcare :-

→ Medical imaging : CV helps in anyalzing medical scans to detect diseases such as X-rays, MRIs & CT scans.

→ Surgical assistance : Used for robotic surgeries for precision & real time image guidance.

→ Disease detection : AI powered vision systems can detect conditions like diabetic retinopathy & skin cancers through image analysis.

ision

2. Autonomous vehicles :-

→ Object detection :- used to identify pedestrains, traffic signs, obstacles & other vehicles.

→ Lane detection :- Helps in lane-keeping & road following for self driving cars.

→ Collision avoidance :- Cameras & sensors detect object in the vehicles path & take action to avoid accidents.

3. Security & Surveillance :-

→ Facial recognition :- Used in security systems for identify verification & access control.

→ Activity monitority :- CV can analyze video feeds to detect suspicious activities, intrusions, or abnormal behaviors.

→ Crowd monitoring :- Deployed at public events or places for crowd control & safety measures.

4. Retail :-

→ Automated checkout :- Stores like amazon Go use CV to track what customers picks up & automatically charge them.

→ Shelf monitoring :- Monitors stock levels & shelf organization alerting staff when products need restocking.

→ Customer Behavior analysis :- Analyzes how customers move through stores, what products they look at & their buying habits.

5. Education :-

→ Smart classrooms :- CV can monitor student enagement & provide feedback to teachers about participation levels.

→ Language learning :- Used in apps to identify & correct pronounciations through lip-reading technology.

Q5. Define imaging, sampling, and quantization.

Ans Imaging → refers to the process of capturing visual information using a sensor like a camera & converting it into a digital image.
The camera captures light reflected from objects in the scene & transforms it into electrical signals. These signals are then converted into pixels, which make up a digital image.
Example → A smartphone camera captures a photo of a landscape, turning the scene into a grid of pixels that represent the image digitally.

Sampling → is the process of converting a continuous signals into a discrete signal by dividing it into individual pixels or sample points.
In the real world, images are continuous range of meaning there are infinite points of information. Sampling selects specific points (pixels) from this continuous scene to represent it digitally. The number of pixels determines the image quality - higher sampling results in a more detailed image.
Example → In digital image, a camera sensor samples the light at regular intervals, capturing a finite number of points to form a grid of pixels e.g (1920 x 1080 pixel image).

Quantization → is the process of mapping the continuous range of pixel values into discrete levels to store the image digitally.

In real life, light & color intensities are continuous, meaning they can take any value within a range. Quantization converts these continuous intensities into discrete values that can be stored as digital numbers. The more levels of quantization, the more accurately the image can represent colors & shades.
Example :- In an 8 bit grayscale image, the continuous range of brightness values is quantized into 256 discrete levels (0 to 255), where 0 represent black and 255 represents white.

Q6 Define neighbours, connected components, connected sets, regions & boundary.

Ans Neighbors → refers to the pixels that are directly adjacent to a given pixel in an image, based on their spatial proximity.
For a given pixel, its neighbors are those surrounding it & the type of connectivity determines which pixels are considered neighbors.
4-neighborhood - 4 pixels - above, below, left & right of pixel.
8-neighborhood - 8 pixels - surrounding pixels including diagnols.
Example → In a grid, if you are focusing on pixel (x, y), the 4-neighbor are (x-1, y), (x+1, y), (x, y-1), & (x, y+1) while the 8 neighbors also include diagonal like (x-1, y-1), (x+1, y+1) etc

2. **Connected Component** - is a group of pixels in an image that are connected to each other based on a defined neighbohood (eg. 4 connected or 8-connected) & share similar properties like intensity or color. Pixels that are neighbours & have similar values are grouped into connected components. This helps in identifying distinct objects or regions in an image. Example - In an binary image (0,1), a connected component would be a group of adjacent '1' pixels that forms a single object.

3. **Connected set** - is a collection of pixels from a continuous group where each pixel can be reached from another pixel in the set through a sequence of neighbor pixels. All pixels in a connected set are part of the same region because they are connected based on the defined neighborhood connectivity. Example :- In a grayscale or binary image, a connected set could be a shape or object where every pixel within the object can be linked to another pixel through adjacent pixels.

4. **Region** - is a subset of pixel within an image that share a specific property, such as similar intensity, color or texture, & are spatially connected. A region can be thought of as an area or object in the image that stands out from its surroundings. Regions can be formed by grouping connected components based on their similarities.

Example - In a segmented image, each region might represent a distinct object or part of an object.

5. Boundary - of a region is the set of pixels that separate the region from its surrounding pixels, marking the edge of the region.
Boundary defines the limit of a region & help in distinguishing it from other regions or the background. Boundary pixels are typically adjacent to pixels that are not part of the region.
Example :- In an object detection task, the boundary would be the outline of the detected object, which differentiates it from the background or other objects.

Q7. What are the different distance measures, explain with formula.

Ans7 1) Euclidean Distance - is the most common & straightforward distance measure, representing the straight-line distance between two point in a multi dimensional space.
Formula :-

$$d(p,q) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \dots + (p_n-q_n)^2}$$

Example - In a 2D image, the Euclidean distance between two pixels at positions (x_1,y_1) & (x_2,y_2) is

$$d = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

2. Manhattan distance (City Block Distance):
The manhattan distance measures the distance between two points along the grid lines, like how you would travel through city blocks.

Formula -

$$d(p,q) = |p_1 - q_1| + |p_2 - q_2| + \dots + |p_n - q_n|$$

Example -
For two pixels at (x_1, y_1) and (x_2, y_2), the distance is

$$d = |x_2 - x_1| + |y_2 - y_1|$$

* Often used in grid based applications like path finding & image segmentation.

3. Cosine similarity (Cosine Distance)
measures the cosine of the angle between two vectors in multidimensional space. Although it is more of a similarly measure; it's often used in CV tasks.

Formula :- $\text{cosine similarity} = \dfrac{p \cdot q}{\|p\| \, \|q\|}$

$$\text{Cosine Distance} = 1 - \text{Cosine similarity}$$

Example -
used to compare two feature vectors of image to see how similar they are in terms of direction, regardless of magnitude

* useful in image comparison to feature extraction where vector direction matters more than magnitude (e.g in text or image similarity).

4. Hamming distance :-
measures the number of positions at which two binary vectors differ.

Formula :- $d(p,q) = \sum_{i=1}^{n} (p_i \oplus q_i)$ $\oplus$ XOR operation

Example :- If p = 1011101 and q = 1001001, the hamming distance is the number of positions where the bits differ (3 in this case).

* Commonly used in error detection & correction, & in comparing binary patterns like in fingerprint matching or object recognition.

Q8. Explain the different point processing operations in the following form:

a Equation of transformation
b Effect
c Graph
d Input image V/s output image.

Ans 1. Negative Transformation :-

a. equation of transformation → $s = L - 1 - r$

s is the output pixel value

r is the input pixel value

L is the number of gray levels

b. Effect - The transformation inverts the grayscale image, turning light areas into dark ones & vice-versa.

c. Graph - The graph of input r v.s. output s is a straight line with a negative slope, starting from L-1 (white) to 0 (black).

ision

2. Logarithmic Transformation

a. Equation of transformation : $s = c \cdot \log(1+r)$

s is the output pixel value

r is the input pixel value

c is a constant, usually to scale the output.

b. Effect :- Transformation enhances the dark regions of the image, allowing more details to be visible in areas that were previously to dark.

c. Graph :- The graph of r vs. s is a curve that increase rapidly for lower values of r and flattens for higher values.

3. Power-Law (Gamma) Transformation

a. equation of transformation : $s = c \cdot r^{\gamma}$

s is the output pixel value

r is the input pixel value

γ is the gamma value (controls degree of transformation)

c is a scaling constant.

b. Effect - Transformation can either lighten or darken an image, depending on the value of γ, If $\gamma < 1$, the image appears lighter, if $\gamma > 1$, the image appears darker.

c. Graph - The plot shows a curve that varies based on the value of γ: it rises more steeply for values less than 1 and flattens for values greater than 1.

4. Thresholding
a. Equation of Transformation

$$s = \begin{cases} 0 & \text{if } r < T \\ L-1 & \text{if } r \geq T \end{cases}$$ [T is the threshold value]

b. Effect -
Thresholding converts a grayscale image into a binary image by setting pixel values to either maximum (white) or minimum (black) based on the threshold.

c. Graph :-
The plot is a step function, where all values below T map to 0 and those above map to L-1.

5. Intensity Level Slicing
a. Equation of Transformation

$$s = \begin{cases} L-1 & \text{if } r_1 \leq r \leq r_2 \\ 0 & \text{otherwise} \end{cases}$$

b. Effect :-
It highlights certain intensity ranges, making them more prominent while darkening the rest of the image.

c. Graph :-
The graph show a flat line at 0 for all values outside the range $[r_1, r_2]$ and jumps to L-1 within the range.

Q9. What is convolution? Explain the following filters with example & effects :- blurring :-

a) Average filter e) mean filter.
b) min filter
c) median filter
d) max filter

vision

Ans Convolution is a mathematical operation used in image processing & CV to combine two functions. It typically refers to applying a filter to an image to produce a modified output image. The process involves sliding the filter over the image & calculating the weighted sum of the pixel values covered by the filter. This operation is widely used for various image processing tasks such as blurring, sharpening, edge detection & noise reduction.

Filters for blurring.
Blurring is a common operation in image processing used to reduce noise & detail.

1. Average filter :- replaces each pixels value with the average value of the surrounding pixels including itself
Kernel Example :- A 3x3 average kernel look like this

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Effect - This filter smoothens the image, reducing detail & noise.

2. Min filter :- replaces each pixels value with the minimum value of the surrounding pixels.
Kernel Example :- A 3x3 min filter kernel can be represented implicitly.
Effect - This filter helps to reduce noise by removing outliers pixels, often used in scenarios where the darkest pixels need to be preserved.

3. Median filter :- replaces each pixel value with the median value of the surrounding pixels.

Kernel Examples :- median filter does not have a specific Kernel matrix but operates by sorting the values in the kernel area & selecting the median.
Effect :- This filter is excellent for removing salt and peeper noise while preserving edges.

4. Max filter - replaces each pixel's value with the maximum value of the surrounding pixels.
Kernel Examples :- The max filter operates by finding the maximum in the kernel area.
Effect :- This filter can enhance bright features in the image while supressing darker details.

5. Mean filter :- is essentially same as the average filter & is often used interchangeably.
Kernel Examples :

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Effect :- This filter smooths the image by averaging pixel values, helping to reduce noise & details.

10. Gray level slicing, intensity slicing, boxplot slicing, bit plan slicing, histogram equalizer.

Gray-level slicing - is a technique used to highlight specific ranges of pixels intensities in an image allowing for better visualisation of particular features.
In a gray level slicing pixel values within a specified range are set to a higher dimensional value, while all over pixel value are set to a lower intensity.

* This enhances the visibility of the chosen intensity range.
Example → If the specified gray level range is [100, 150], all pixel values between 100 and 150 will be set to high value (e.g. 255), while all over the values will be set to 0.
Effect → The output image emphasizes the specified gray levels, making features within that range stand out.

2. Intensity slicing :- similar to gray level slicing but focuses on range of intensities rather than specific gray levels.
* Intensity slicing segments the image based on intensity values, where a specific intensity range is selected, & all pixels in this range are enhanced.
* This can be visualized in a grayscale image where selected intensity ranges are set to a brighter value.
Example → For intensity values in the range [50, 100], those pixels are enhanced, while others may remain unchanged or be set to a different lower value
Effect → This enhances certain features while maintaining the rest of the image in a less emphasized state, useful in applications like medical imaging.

3. Boxplot slicing :- is a technique that utilizes boxplots to determine intensity levels & helps visualize how pixel intensities are distributed across an image.
* Boxplot slicing identifies the quartiles & outliers in pixel intensity distribution. It can be used to adjust contrast & highlight specific ranges in images.
* By visualizing the pixel distribution, certain ranges can be enhanced or reduced based on statistical metrics

Example :- The boxplot indicates specific intensity ranges to focus on; pixel values corresponding to the upper quartile may be enhanced.
Effect - Enhance visual analysis by providing insight into the distribution of pixel intensities, which can aid in determining important features.

4. Bit-Plane Slicing :- seperates an image into its individual bits, allowing manipulation of specific bits that contributes to the image's intensity values.
* Each pixel's intensity can be represented in binary. For an 8 bit image, each pixel can be divided into 8 layers.
* Each bit plane can then be processed separately.
Example - If you extract the third bit-plane from a pixel value of 10101100, it corresponds to 1, highlighting areas of low intensity in that bit position.
Effect - Allows selective enhancement & analysis of image features, often used in data compression & steganography.

5. Histogram Equalization :- is a technique that enhances the contrast of an image by adjusting intensity distribution.
* Histogram of pixel intensity values is computed and a cumulative distribution function is derived. The CDF is then used to map the original pixel values to new intensity values, spreading them across the full range of possible values.
* The goal is to achieve a uniform histogram, which results in better contrast in the output image.
Example :- In an image with limited intensity ranges, histogram equalization redistributes the pixel value to

utilize the full intensity range (0-255).
Effect:- Results in increased contrast & visibility of features that may not be prominent in the original image, making it particularly useful for improving the visibility of details in dark or bright areas.

Q11. Define an edge. What are the different types of edges? What are the two ways to extract the edges (First order derivative or gradient) or second-order derivative or laplacian).

Ans Edge is a boundary or transaction between two different regions in an image. It typically represents a significant change in intensity or color, corresponding to object boundaries, texture, changes, or illumination variations in a scene.

Types of edges:-

1. Step Edge - Abrupt and significant change in intensity between two adjacent regions
2. Ramp Edge - A more gradual change in intensity over a finite distance.
3. Line Edge - Characterized by a narrow, strong transition in intensity in a specific location.
4. Roof Edge - Similar to a line edge, but with a smooth, rounded peak
5. Junctions / Corners - Points where two or more edges meet.

Methods to Extract Edges

1. First-order Derivative (Gradient Based) Methods:- These method detects edges by calculating the gradient of the image intensity function. The gradient gives information about the rate of change of intensity, helping to locate

edges where there are significant intensity changes.

→ Sobel operator :- This operator computes the gradient of the image intensity in both the horizont (G_x) and vertical (G_y) directions using convolution masks. The magnitude of the gradient is used to find edges:

$$G = \sqrt{G_x^2 + G_y^2}$$

→ Prewitt operator :- Similar to Sobel but uses simpler masks to approximate the gradient.

→ Canny edge operator :- A multi-step algorithm that uses Gaussian smoothing, gradient calculation, non maximum suppression, and double thresholding to find edges.

2. Second-Order Derivative (Laplacian-Based) Method:

These methods use the second derivative to find areas where the rate of change of intensity changes sharply. A common approach is the Laplacian of Gaussian (LOG), where the image is first smoothed with a Gaussian filter and then the Laplacian is applied. Zero-crossings in the second derivative indicate the presence of edges.

→ Laplacian operator : It computes the second derivative of the image intensity, identifying regions where intensity changes sharply.

$$\Delta^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

A positive-to-negative or negative-to-positive transition in the second derivative indicates an edge.

→ Difference of Gaussians (DoG) : This method approximates the LOG by substracting two Gaussian-blurred versions of the image.

Q12. Write down the filter & masks for Roberts, Sobel, Prewitt and Laplacian operators.

Ans 12- 1) Roberts Operator: is a simple gradient operator that detects edges by calculating the difference between diagonal pixels.

Robert X (Gx) : $\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$

Robert Y (Gy) : $\begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$

2) Sobel operator: is an improvement over the Prewitt operator as it introduces a smoothing effect by giving more weight to the center pixels. It is widely used for gradient calculation.

Sobel X (Gx) : $\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$

Sobel Y (Gy) : $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$

3) Prewitt operator: is similar to Sobel but simpler as it doesn't assign different weights to the center pixels. It approximates the gradient of the image.

Prewitt X (Gx) : $\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$

Prewitt Y (Gy) : $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$

4. Laplacian Operator : is used for detecting regions of rapid intensity change & highlights areas of edges by calculating the second-order derivative. It is isotropic, meaning it detects edges in all directions.

→ 4-neighborhood Laplacian :

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & +4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

→ 8-neighborhood Laplacian (used for more precision)

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Q13. Explain the following image transformation & their effects - translation, rotation, shear, reflection.

Ans Image transformation are operations that modify an image's geometry by changing its pixel locations.

1. Translation - is the shifting of an image by moving all its pixels by a certain distance in x & y directions

→ Effect :- The entire image is displaced horizontally (x-direct) vertically (y-direction), or both without any change in its orientation, shape, or size. Parts of the image that are translated out of the frame are lost, & new areas (gaps) may be filled with a default background color.

→ Transformation Matrix :

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{bmatrix}$$

where tx and ty are the translation distance along the x & y axis.

- Effect in practice : If you move an object in an image to the right by 50 pixels & down by 30 pixels, the object appear at a new location, but its orientation & appearance remain unchanged.

2. Rotation :- involves rotating an image around a fixed point, usually the centre of the image, by a certain angle in clockwise or counterclockwise direction.

-> Effect :- The image rotates around the center, changing the orientation of the objects in the image. The size & shape of the objects remain unchanged, but the parts of the image may get cut off depending on the angle of rotation, & blank areas may appear at the corners.

-> Transformation matrix : where θ is the angle of rotation

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-> Effect in practice - If you rotate an image by 45°, objects in the image will appear turned by that angle. Clockwise rotation results in a negative angle, while counterclockwise result in a positive angle.

3. Shear : is a transformation that skews the image in one direction, either horizontally or vertically. It distorts the image by shifting one axis, making the image "slant" without preserving angles between lines.

-> Effect :- The image appears stretched or compressed in one direction, causing a parallelogram-like distortion. For ex- vertical lines may tilt to the side or horizontal lines may stretch at an angle, changing the shape of objects.

-> Transformation Matrix (Horizontal shear):

$$\begin{bmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where λ is the shear factor, controlling the amount of distortion.

→ Transformation Matrix (Vertical shear):

$$\begin{bmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where λ is the shear factor

→ effect in practice: In horizontal shear, objects appear slanted horizontally. In vertical shear, objects get skewed vertically, distorting their shapes.

4. Reflection: creates a mirror image of the original, flipping the image either across a horizontal, vertical, or diagonal axis.

→ Effect :- The image is flipped across the specified axis, creating a mirror version of itself. In a horizontal reflection, the top and bottom parts of the image are swapped, whereas in a vertical reflection, the left & right parts are swapped. Diagonal reflections flip both axes.

→ Transformation matrix (Horizontal Reflection):

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix flips the image across the horizontal axis.

→ Transformation matrix (Vertical Reflection):

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix flips the image across the vertical axis

→ Effect in pratice :- A vertical reflection mirrors the image left-to-right, while a horizontal reflection mirrors it top-to bottom. The shape size & orientation are preserved, but the visual direction changes.

VISION

Q14 Write the matrices in 2D and 3D. What is the advantage of using 3x3 matrix for transformation.

Ans Transformation Matrices in 2D

1. Translation (2D): moves an image by given amount along x & y axis.

Translation Matrix (2D) = $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$

t_x = translation in x direction

t_y = translation in y direction

2. Rotation (2D): turns the image around the origin by an angle θ.

Rotation matrix (2D) = $\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. Shear (2D): skews the image along one axis.

* Horizontal shear = $\begin{bmatrix} 1 & \lambda & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

λ is the shear factor in x direction.

* Vertical shear = $\begin{bmatrix} 1 & 0 & 0 \\ \lambda & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

λ is the shear factor in y direction.

4. Reflection (2D): flips the image along a specific axis.

* Reflection matrix = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Reflects across the x axis
" Horizontal reflection "

* Vertical reflection: $\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Reflects across the y-axis.