Name: Prathamesh Arvind Jadhav

UNIT-3

Representation in Computer Vision

Introduction

In computer vision, **representation** refers to the way an image or an object is described and stored for further processing and analysis. Since computers do not "see" images like humans, they rely on numerical representations that capture essential information about an object, such as its shape, color, texture, and structure.

Representation plays a crucial role in **object detection, recognition, classification, and segmentation**, enabling machines to understand and process visual data efficiently.

Types of Representation in Computer Vision

There are two primary types of representation in computer vision:

1. **Iconic Representation (Raw Data Representation)**
2. **Symbolic Representation (Feature-Based Representation)**

1. Iconic Representation (Raw Data Representation)

This type of representation preserves the original structure of the image. It contains **pixel-level information** without any abstraction.

Forms of Iconic Representation:

1. **Pixel Representation (Raster Images)**
   - The image is stored as a matrix of pixel intensity values.
   - Example: **Grayscale images (0-255 intensity levels)** or **RGB images (three color channels - Red, Green, Blue).**
2. **Binary Representation (Black and White)**
   - An image is converted into a binary format where each pixel is either **0 (black)** or **1 (white).**
   - Useful for **edge detection, OCR (Optical Character Recognition), and segmentation tasks.**

3. **Vector Representation**
    - Stores images as a set of mathematical **curves and shapes** rather than pixels.
    - Common in **graphics design, CAD (Computer-Aided Design), and shape analysis.**

Example of Iconic Representation:

- **Handwritten Digit Recognition**: The MNIST dataset consists of grayscale images of handwritten digits, where each digit is represented as a **28×28 pixel matrix.**

---

2. Symbolic Representation (Feature-Based Representation)

Symbolic representation converts an image into a **set of features** that describe its shape, texture, or other attributes rather than storing pixel-level details.

Forms of Symbolic Representation:

1. **Shape-Based Representation**
    - Objects are represented using their **boundaries, contours, or skeletons.**
    - Example: **Contour extraction using OpenCV's findContours() function.**
2. **Feature-Based Representation**
    - Instead of using raw pixel values, images are represented using **key features** like corners, edges, and textures.
    - Examples:
        - **Histogram of Oriented Gradients (HOG):** Used for **pedestrian detection.**
        - **SIFT (Scale-Invariant Feature Transform):** Used for **object recognition.**
3. **Graph-Based Representation**
    - Objects are represented as a **graph of interconnected nodes**, where nodes represent key points, and edges define relationships.
    - Example: **Skeleton representation of human pose estimation.**
4. **Model-Based Representation**
    - Uses **3D models** to represent objects.
    - Example: **3D reconstruction in AR/VR applications.**

Name: Prathamesh Arvind Jadhav

Importance of Representation in Computer Vision

✔ **Efficient Storage** – Reduces memory usage by extracting only essential features.
✔ **Faster Processing** – Symbolic representations allow quicker computation for real-time applications.
✔ **Improved Recognition** – Helps in **classification, object detection, and segmentation.**
✔ **Robustness** – Feature-based representations are more robust against **scaling, rotation, and noise.**

Boundary Descriptor in Computer Vision

Introduction

A **boundary descriptor** is a feature extraction method in computer vision that focuses on the **shape and contour** of an object in an image. Instead of analyzing the entire region of an object, boundary descriptors only consider the **outline** or **edges**, which makes them useful for **shape recognition, object detection, and classification**.

Boundary descriptors play a crucial role in applications like **signature verification, fingerprint recognition, and optical character recognition (OCR)**.

Importance of Boundary Descriptors

✔ **Compact Representation** – Instead of processing the entire object, only the boundary is stored.
✔ **Rotation and Scale Invariance** – Many descriptors provide a shape representation that is independent of rotation and size.
✔ **Efficient Matching** – Used in shape-based object recognition to compare similar shapes.
✔ **Noise Reduction** – Reduces unwanted variations in the object's interior.

Name: Prathamesh Arvind Jadhav

Types of Boundary Descriptors

Boundary descriptors can be categorized into two main types:

1. **Statistical Descriptors** – Based on the numerical properties of the contour.
2. **Structural Descriptors** – Based on the geometric or topological features of the boundary.

1. Statistical Boundary Descriptors

These descriptors represent the boundary using mathematical properties such as **length, curvature, Fourier descriptors, and moments**.

(a) Perimeter (Contour Length)

- Measures the total length of the object's boundary.
- Computed using the sum of distances between consecutive points on the contour.

- Formula:

$$P = \sum_{i=1}^{N} d(p_i, p_{i+1})$$

where $d(p_i, p_{i+1})$ is the Euclidean distance between consecutive boundary points.

(b) Curvature-Based Descriptor

- Represents how the contour bends along its shape.
- Defined as the **rate of change of the tangent angle**.
- Used in **fingerprint recognition and signature verification**.

(c) Fourier Descriptors

- Uses **Fourier Transform** to represent a contour in the frequency domain.
- Extracts important shape features while ignoring small variations.
- Helps in **scale, translation, and rotation-invariant shape matching**.

(d) Shape Signature (Centroid Distance Function)

- Represents the **distance of each boundary point from the centroid**.
- Helps in detecting similar shapes even if they are **rotated or scaled**.

---

2. Structural Boundary Descriptors

These descriptors focus on **the structure of the boundary**, such as the number of corners, edges, or inflection points.

(a) Chain Code Representation

- Represents a boundary as a sequence of direction codes.
- Example: **Freeman Chain Code**, which uses 8 directions (0-7) to represent movement from one boundary point to another.

(b) Polygon Approximation

- Simplifies a shape by approximating its boundary with straight-line segments.
- Example: **Ramer-Douglas-Peucker algorithm** reduces unnecessary points while keeping the shape intact.

(c) Skeleton Representation

- Reduces an object to its central skeleton by **thinning** the boundary.
- Used in **handwriting recognition and medical image analysis**.

Applications of Boundary Descriptors

- **Signature and Handwriting Recognition** – Uses shape descriptors to verify authenticity.
- **Medical Image Analysis** – Extracts shape-based features for tumor detection.
- **Shape Matching and Object Detection** – Identifies similar objects based on contours.
- **Fingerprint and Biometric Recognition** – Analyzes boundary structures for identification.
- **Character Recognition (OCR)** – Recognizes characters based on their boundary structure.

Name: Prathamesh Arvind Jadhav

Regional Descriptor in Computer Vision

Introduction

A **regional descriptor** is a feature extraction method in computer vision that focuses on the **entire region of an object** rather than just its boundary. It captures **texture, intensity, shape, and statistical properties** inside an object, making it useful for **image classification, segmentation, and object recognition**.

Regional descriptors are particularly helpful when objects have complex or **irregular boundaries** but distinctive **internal patterns**.

Importance of Regional Descriptors

✔ **Comprehensive Representation** – Considers both the shape and texture of an object.
✔ **Robust Against Noise** – Less sensitive to small variations in object edges.
✔ **Useful for Texture Analysis** – Helps distinguish objects based on texture rather than shape.
✔ **Essential for Medical Imaging** – Used in tumor detection and tissue classification.

Types of Regional Descriptors

Regional descriptors can be categorized into:

1. **Statistical Descriptors** – Describe the distribution of intensity values in a region.
2. **Structural Descriptors** – Capture spatial relationships within a region.

1. Statistical Regional Descriptors

These descriptors use numerical properties such as **mean, variance, moments, and histograms** to describe an object's internal structure.

(a) Area (Region Size)

- Measures the number of pixels inside an object.
- Used in **segmentation and shape classification**.

(b) Moment-Based Descriptors

- Compute statistical moments to describe object properties.
- Example: **Hu Moments**, which are **invariant to rotation, translation, and scaling**.

  - Formula for second-order moments:

    $$M_{pq} = \sum_x \sum_y x^p y^q f(x,y)$$

    where $f(x,y)$ is the pixel intensity at (x,y).

(c) Histogram-Based Descriptors

- Capture intensity variations in a region.
- Example: **Color histograms** in **RGB or HSV** space.

(d) Texture Descriptors (Gray Level Co-occurrence Matrix - GLCM)

- Measures **how frequently pixel intensities appear together** in a region.
- Used in **texture classification (e.g., fabric, skin patterns, satellite images)**.

---

2. Structural Regional Descriptors

These descriptors capture the spatial organization of pixels.

(a) Local Binary Pattern (LBP)

- Assigns a binary code to each pixel based on its neighboring pixel intensities.
- Used in **face recognition and texture classification**.

(b) Histogram of Oriented Gradients (HOG)

- Captures the **gradient direction** and **magnitude** of edges in a region.

- Used in **pedestrian detection, object recognition**.

(c) Zernike Moments

- Represent a region using orthogonal polynomials.
- Used in **medical imaging and shape recognition**.

Applications of Regional Descriptors

- **Medical Imaging (Tumor Detection)** – Analyzes region texture to classify tissues.
- **Face Recognition** – Uses **LBP and HOG** for feature extraction.
- **Satellite Image Analysis** – Identifies land types based on texture.
- **Object Classification** – Helps in **AI-based object detection** in smart cameras.
- **Defect Detection in Manufacturing** – Identifies texture differences in industrial components.

Principal Component Analysis (PCA) in Computer Vision for Feature Description

Introduction

Principal Component Analysis (PCA) is a **dimensionality reduction** technique used in **computer vision** to extract essential features from images while reducing noise and redundancy. PCA transforms high-dimensional data into a lower-dimensional space while **preserving important information**.

Why PCA in Computer Vision?

✔ **Reduces Computational Cost** – Fewer features mean faster processing.
✔ **Removes Redundant Information** – Eliminates correlated or less important features.
✔ **Improves Classification Performance** – Enhances the accuracy of machine learning models by focusing on essential features.
✔ **Noise Reduction** – Removes insignificant variations in data.
✔ **Visualization** – Projects high-dimensional data into 2D or 3D space for analysis.

Name: Prathamesh Arvind Jadhav

Applications of PCA in Computer Vision

1. Face Recognition (Eigenfaces)

- PCA is widely used in **face recognition systems** (e.g., Eigenfaces).
- Converts high-dimensional face images into a **low-dimensional space** while keeping the essential identity information.

2. Object Detection

- PCA extracts **distinctive features** from an object, making it easier to classify objects in cluttered backgrounds.
- Used in **vehicle detection, human pose estimation**, etc.

3. Image Compression

- PCA helps **compress images** by keeping only the most significant features.
- Reduces storage and computational requirements while maintaining image quality.

4. Medical Image Analysis

- PCA is used in **MRI and X-ray image processing** to highlight important structures.
- Helps in detecting **tumors, fractures, or abnormalities**.

5. Gesture and Action Recognition

- PCA reduces the dimensionality of motion data in **human activity recognition**.
- Used in **gesture-based interfaces, sign language recognition**, etc.

Advantages of PCA in Feature Description

Feature	PCA Benefit
Dimensionality Reduction	Reduces complexity while retaining key features
Noise Filtering	Eliminates irrelevant variations

Name: Prathamesh Arvind Jadhav

Feature	PCA Benefit
Faster Processing	Reduces computation time in ML models
Better Generalization	Improves performance of classifiers

Mathematics Behind PCA

PCA works by **finding the directions (principal components) that capture the maximum variance in the data.** The steps involved are:

1. Compute the Mean and Center the Data

Given an image dataset X of size $m \times n$ (where m is the number of images and n is the number of features per image), subtract the mean from each feature:

$$X_{\text{centered}} = X - \text{mean}(X)$$

2. Compute the Covariance Matrix

The covariance matrix measures how different features vary together:

$$C = \frac{1}{m} X_{\text{centered}}^T X_{\text{centered}}$$

3. Compute Eigenvalues and Eigenvectors

Solve for the eigenvalues and eigenvectors of the covariance matrix:

$$Cv = \lambda v$$

where:

- v represents **principal components (eigenvectors)**
- λ represents **variance along those components (eigenvalues)**

4. Select Top K Principal Components

Sort eigenvalues in descending order and select the **top k eigenvectors** to form a new feature space.

5. Transform Data into Lower-Dimensional Space

Project the original data onto the new basis:

$$X_{\text{reduced}} = X_{\text{centered}} V_k$$

where V_k is the matrix containing the top k eigenvectors.