Module 6 : Pattern Recognition

- Pattern Recognition fundamentals

1. What is a Pattern?

- A pattern is an abstraction represented by a set of measurements describing a physical object, process, or event.
- Patterns can take various forms, including visual, temporal, sonic, and logical.

2. Pattern Recognition Basics

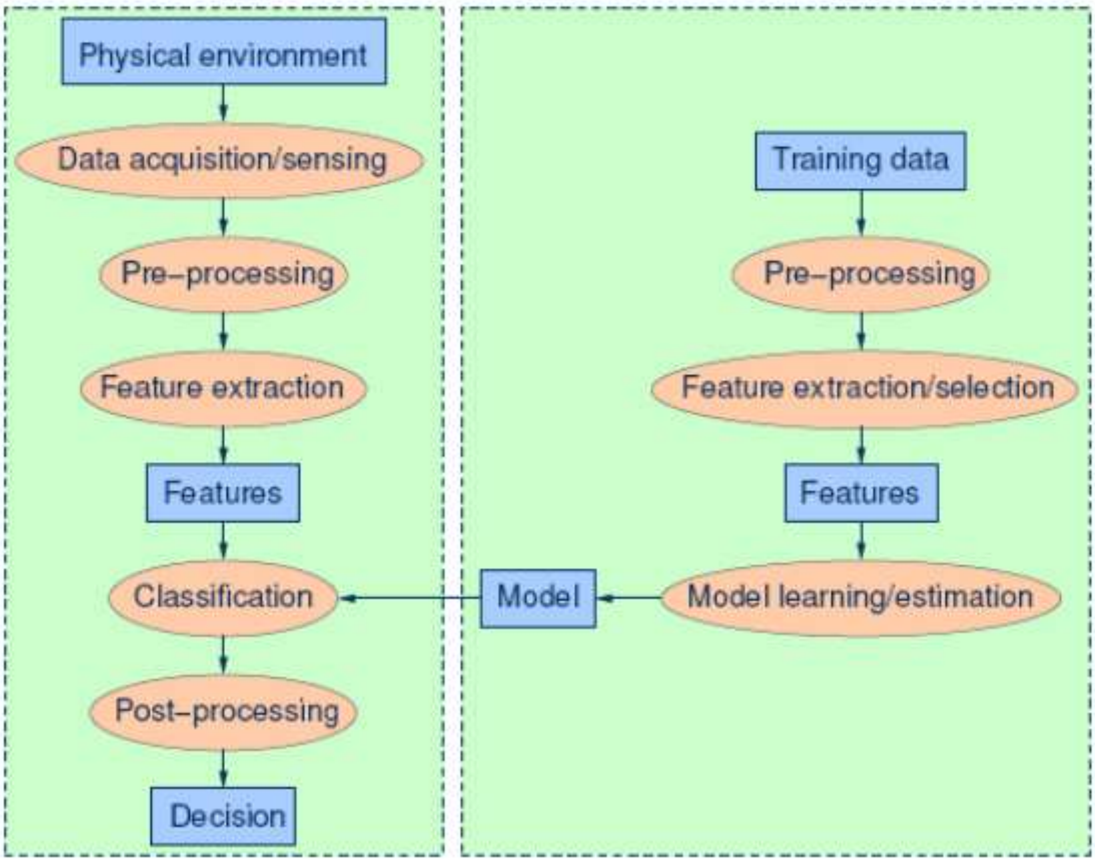
- A pattern class (or category) is a set of patterns that share common attributes and typically originate from the same source.
- During recognition or classification, objects are assigned to predefined classes based on their attributes.
- A classifier is a machine or algorithm that performs classification by assigning patterns to specific classes.

3. Approaches to Pattern Recognition

- **Statistical PR:** Based on underlying statistical models of patterns and pattern classes.
- **Neural Networks:** Classifiers are represented as networks of cells modeling neurons of the human brain (connectionist approach).
- **Structural (or Syntactic) PR:** Pattern classes are represented by formal structures such as grammars, automata, or strings.

Imp

4. Basic Components of a Pattern Recognition System



- **Data Acquisition and Sensing:** Collecting data from the environment, often involving sensors.
- **Pre-processing:** Removing noise from data and isolating patterns of interest from the background.
- **Feature Extraction:** Finding a new representation of patterns in terms of features, which are better suited for further processing.
- **Model Learning and Estimation:** Learning a mapping between features and pattern groups.
- **Classification:** Using learned models to assign patterns to predefined categories.
- **Post-processing:** Evaluating confidence in decisions and exploiting context to improve performance.

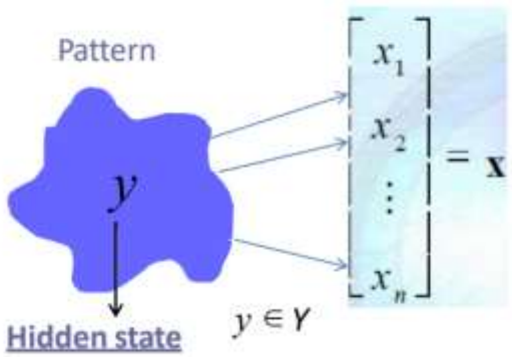
5. Pattern Representation

- A pattern is represented by a set of features or attributes, viewed as a d-dimensional feature vector: $T\mathbf{x}=(x1,x2,\ldots,xd)T$.
- Each feature provides information about a specific aspect of the pattern, allowing for its characterization and classification.

➢ Feature extraction:

1. What is a Feature?

- A feature is a scalar value that quantitatively describes a property of an object, process, or event.
- Features are crucial for distinguishing between different classes of patterns.



Feature Vector

- A feature vector **x** represents a point in the feature space X.
- It comprises observations or measurements denoted by x.

Hidden State

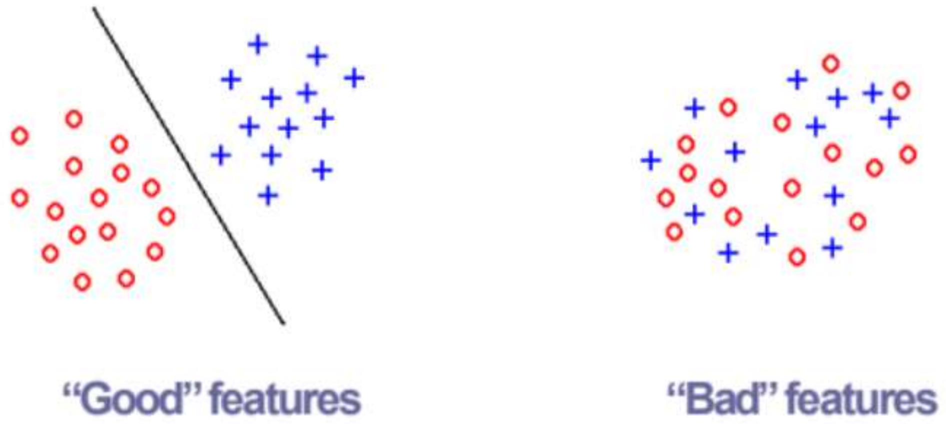
- The hidden state y cannot be directly measured.
- Patterns with the same hidden state belong to the same class.
- It represents the underlying classification or categorization of patterns.

Task

- The task in pattern recognition is to design a classifier, also known as a decision rule

2. Feature Extraction Basics

- Feature extraction involves **selecting and designing** features that are effective for preserving class separability.

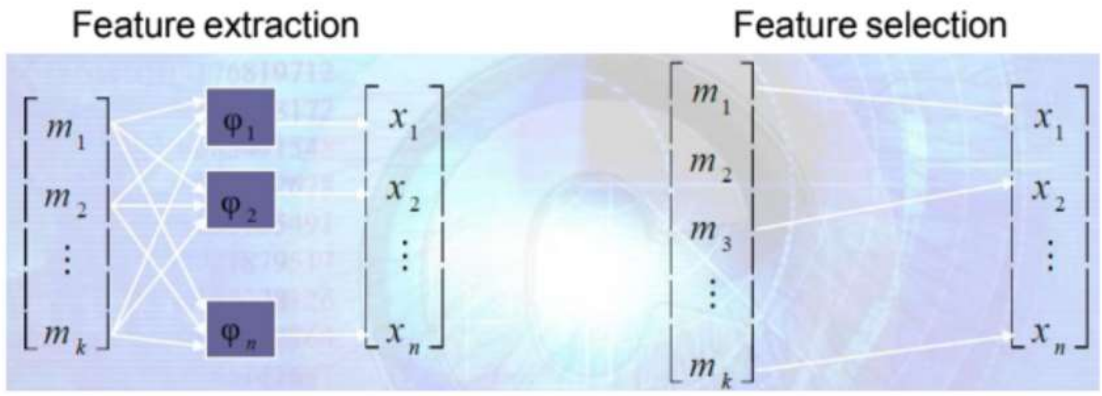


- Good features exhibit the following characteristics:
  - *Objects from the same class have similar feature values*.
  - *Objects from different classes have different feature values.*

3. Methods of Feature Extraction

- **Feature extraction:** Finding a new representation of patterns in terms of features.

- **Feature selection:** Selecting the most relevant features for classification.



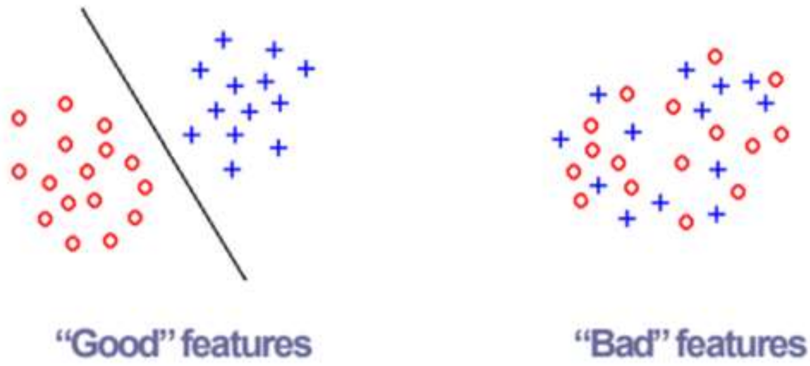
4. Feature Extraction Techniques

- **Supervised Methods:** Objective function is based on the separability or discriminability of labeled examples. Example: Linear Discriminant Analysis (LDA).

- **Unsupervised Methods:** Seek lower-dimensional representations that preserve important characteristics of the input data. Example: Principal Component Analysis (PCA).

5. Challenges in Feature Extraction

- **Inadequate Features:** Features may lack the information needed to separate classes effectively. *Solution:* go back and design better features.



- **Correlated Features:** Features designed to measure different characteristics may be influenced by common mechanisms, leading to correlation.

    - This can degrade classifier performance based on Euclidean distance or other metrics.
    - Solutions include using alternative metrics (e.g., Mahalanobis distance) or extracting uncorrelated features.

6. Importance of Feature Design

- **Problem-Specific Design:** Feature extractors are designed based on the problem domain. For example, features for graphic objects may differ from those for sound events.

- **Ideal Feature Extractor:** Produces the same feature vector for all patterns in the same class and different feature vectors for patterns in different classes.
- **Art and Science:** Designing a good set of features is often considered more of an art than a science, requiring careful consideration of measurement costs, noise, and correlations.

7. Considerations in Feature Extraction

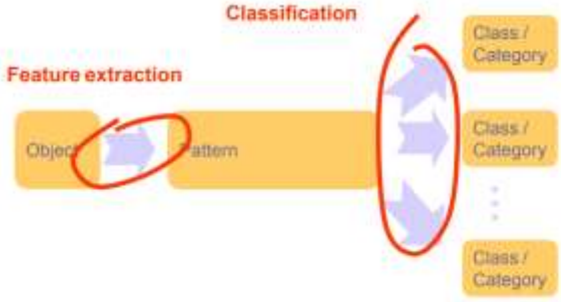
- **Multiple Features:** Adding more features does not always improve results; unreliable features should be avoided, and correlations with existing features should be carefully considered.
- **Curse of Dimensionality:** Working in very high dimensions can lead to challenges, such as the curse of dimensionality, where the number of examples in the training dataset needs to be significantly larger than the number of features.

➢ Classifier:

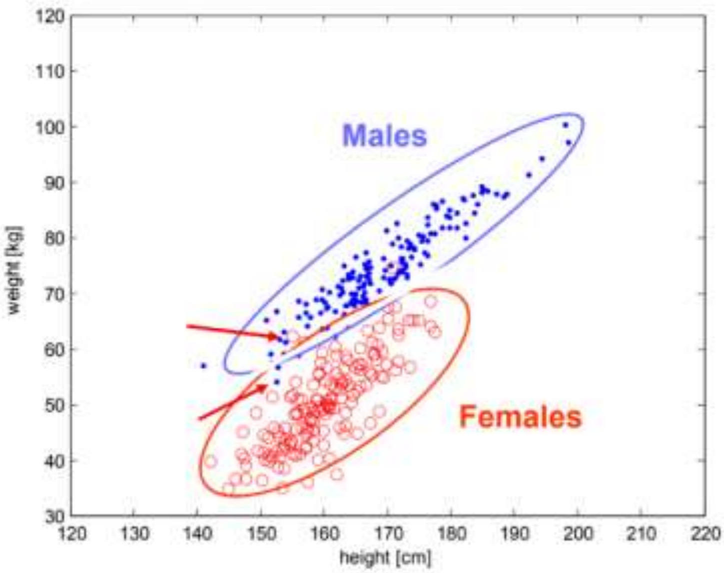
Classifier in Pattern Recognition

- **Class Definition:**
  - A class is a set of patterns sharing common properties. For example, male and female students in a university represent two classes based on their common property of gender.

- **Classification:**
  - Classification **is the process of assigning a feature vector to one of the predefined classes using a mathematical function or algorithm**.



- It involves determining the hidden state (class) based on observed features.
- In the context of the example, a line can be drawn between clusters of male and female students, and each student can be classified as male or female based on this line.

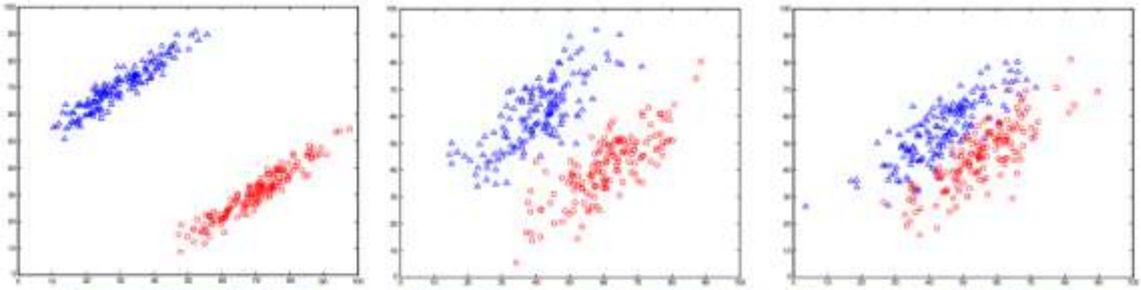


Clusters Separation

Misclassifications are a consequence of the separation of the clusters.

The separation of clusters is quantified using two major methods:

1. Mathematically: there are several *separation criteria*'s.
2. "Intuitively": overlapping of the clusters.



Separable clusters | Almost separable clusters | Non-separable clusters

Imp

Designing a Classifier:

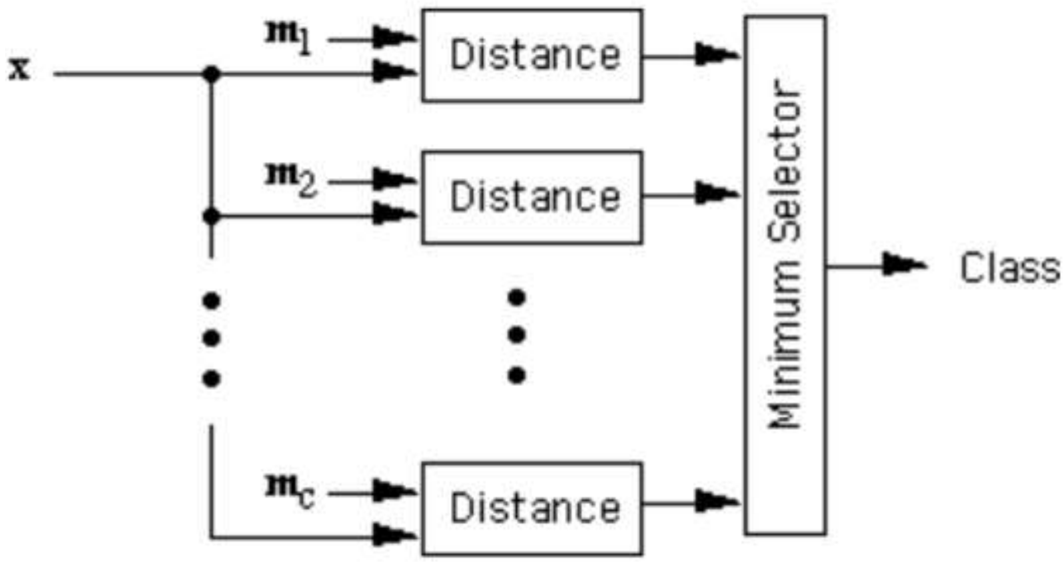
1. **Model Selection:**
    - Depends on domain-specific knowledge and prior information.
    - Consideration of *design criteria, parametric vs. non-parametric models, handling missing features, and computational complexity*.
2. **Complexity vs. Performance:**
    - Trade-off between the *complexity of decision rules and their performance on unknown samples*.
    - Different criteria may lead to different decision boundaries.
3. **Curved Boundaries:**
    - Linear boundaries may not be flexible enough for complex data.

- Solutions include redesigning features, using alternative distance *metrics like Mahalanobis distance, or employing neural networks.*

4. **Subclasses and Clustering:**
   - Classes defined by users may not reflect natural groupings in data.
   - Clustering techniques can help identify subclasses or natural groupings within the dataset.

Imp **Types of Classifiers:**

1. **Simple Classifiers:**
   - Include **minimum-distance classifiers** based on some specified metric.

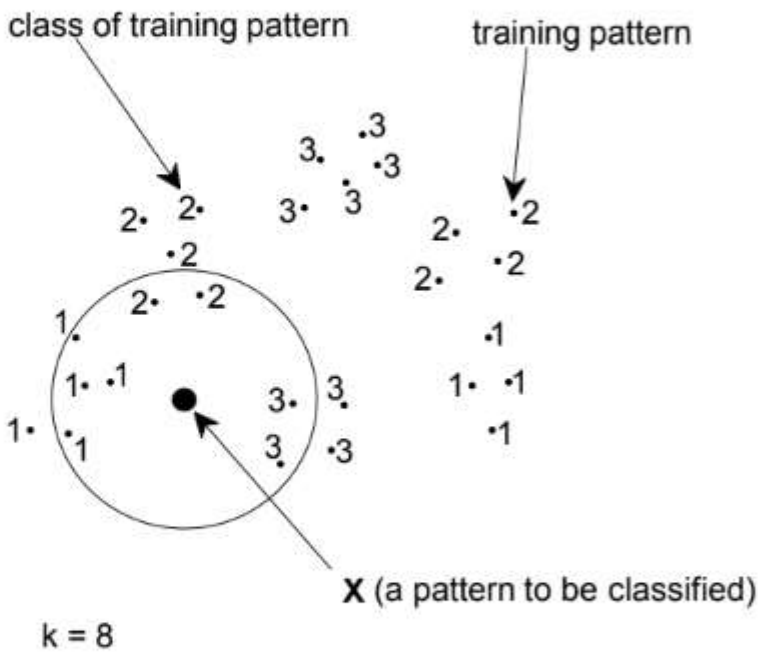


   - Example: Template matching, where patterns are compared to templates using correlation or error-based approaches.

2. **Neural Networks:**

- Represented by interconnected layers of neurons that learn complex relationships between input and output.
- Used for nonlinear classification tasks.

3. **k-Nearest Neighbors (kNN):**
    - Lazy classifier that assigns a class based on the majority vote of its k nearest neighbors.



- No training is performed; classification is based on similarity to labeled examples.

4. **Decision Trees:**
    - Learn rules from data and apply them at each node of a tree structure.
    - Classification is performed at the leaf nodes of the tree based on input features.

Evaluating a Classifier:

1. **Training and Testing Sets:**

- Separate datasets used for training and testing to assess generalization abilities.
- *Avoids overfitting* and provides insight into classifier performance on unseen data.

2. **Confusion Matrix:**
    - Tabulates true positives, false positives, true negatives, and false negatives to evaluate classifier performance.

		prediction outcome		
		p	*n*	total
actual value	*p'*	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
	total	P	N	

3. **Costs of Error:**
    - Consider the consequences of different types of classification errors.
    - For example, misclassifying premium products as standard may incur higher costs than vice versa.

➢ Enlist and Explain any one application of of Pattern Recognition

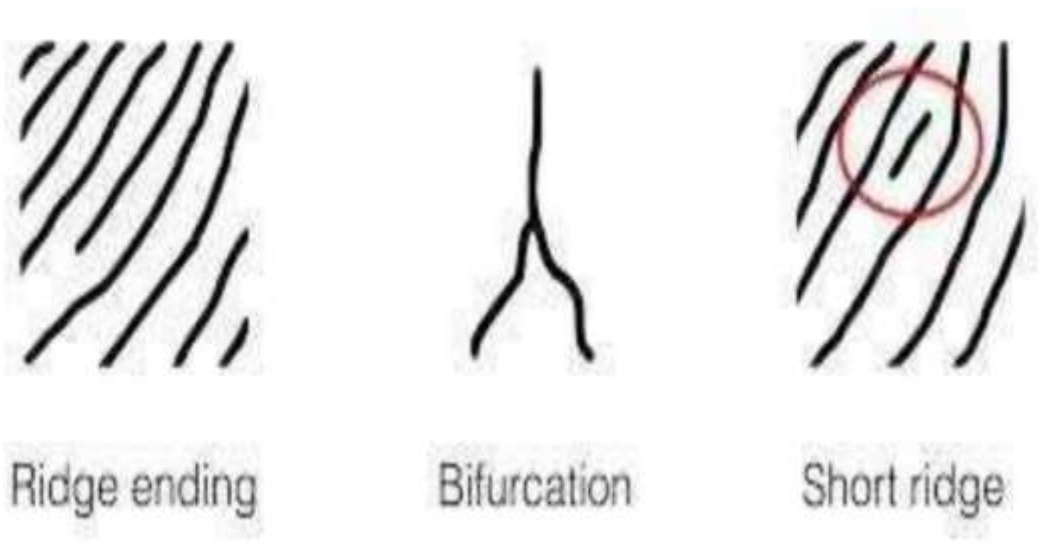
v. Imp

1. - **Biometrics:**
    - Fingerprint recognition, iris recognition, face recognition, and voice recognition
2. **Medical Imaging:**
3. **Speech Recognition:**

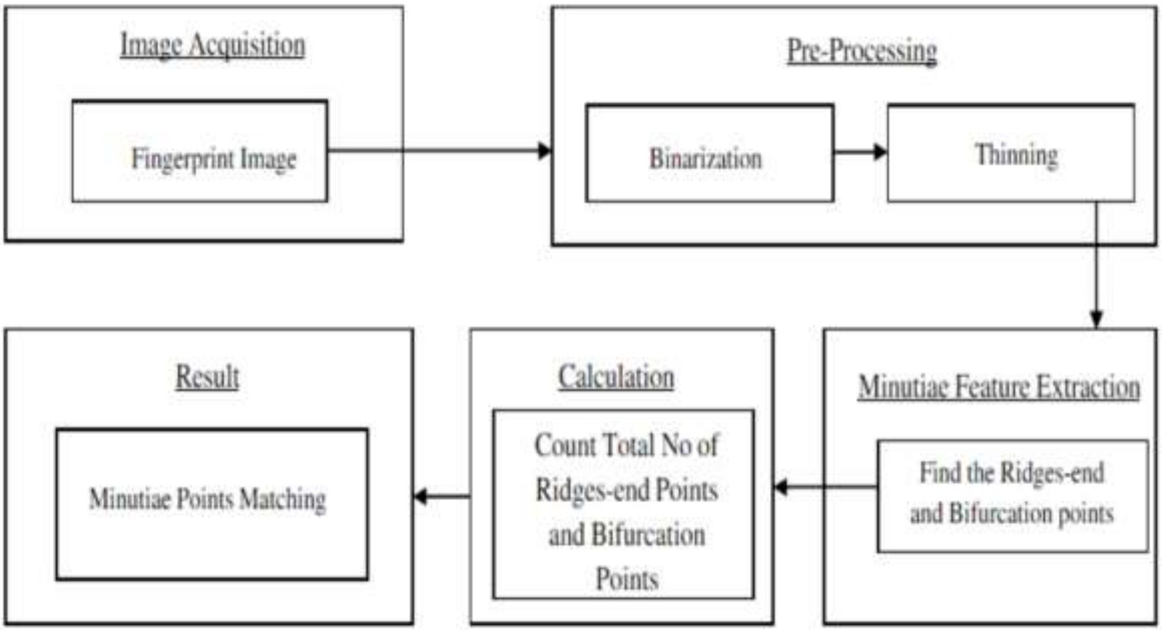
4. **Natural Language Processing (NLP):**
5. **Computer Vision:**
6. **Financial Fraud Detection:**
7. **Gesture Recognition**
8. **Predictive Maintenance:**
9. **Retail Analytics**
10. **Pattern Recognition in Astronomy:**

Explanation of Fingerprint recognition :

- **Fingerprint Patterns:**
  - Fingerprint patterns consist of ridges and valleys, with ridges having characteristic minutiae such as ridge endings and ridge bifurcations.
  - Ridge endings occur where ridges abruptly end, while ridge bifurcations occur where a ridge forks into branch ridges.



- **Fingerprint Recognition:**



- Fingerprint recognition verifies a match between two human fingerprints.
- It offers the advantage of using low-cost standard capturing devices, although recognizing fingerprints can be challenging, especially with noisy or low-quality images.

- **Acquisition of Fingerprint Image:**
  - Fingerprint images are typically **grayscale images with intensity values ranging from 0 to 255**.
  - Various methods, such as inked impression or inkless scanners, are used to acquire fingerprints.

- **Binarization:**
  - Binarization converts the grayscale fingerprint image into a binary image by applying a threshold value.
  - Pixels with values above the threshold are set to '1', while those below are set to '0'.

- **Thinning:**

- Thinning reduces the thickness of ridge lines in the binarized image to a single-pixel width using block filters.
- This process aims to extract minutiae points effectively while preserving their original locations.

- **Minutiae Extraction:**
  - Minutiae points, including ridge endings and bifurcations, are extracted from the thinned fingerprint image.
  - The crossing number method is commonly used to locate minutiae points based on differences in intensity values between adjacent pixels.

- **Minutiae Matching:**
  - Minutiae matching involves comparing the extracted minutiae points from the input fingerprint image with those from a template image retrieved from a database.
  - The process includes computation of points, location detection of points, and matching based on the number and location of minutiae points.

- **Matching Algorithm:**
  - The matching algorithm computes the total number of ridge-end and bifurcation points in the fingerprint image and stores their locations.
  - During matching, the algorithm compares the computed values with stored data to determine if a match occurs based on the combination and location of minutiae points.