References

- **Pattern Classification (2nd ed)** by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000

- Pattern Recognition and Classification, An Introduction, by Geoff Dougherty, Springer Science Business Media New York 2013
- Image processing and pattern recognition Fundamentals and Techniques by FRANK Y. SHIH, 2010 by the Institute of Electrical and Electronics Engineers.

Pattern recognition

What is a Pattern?

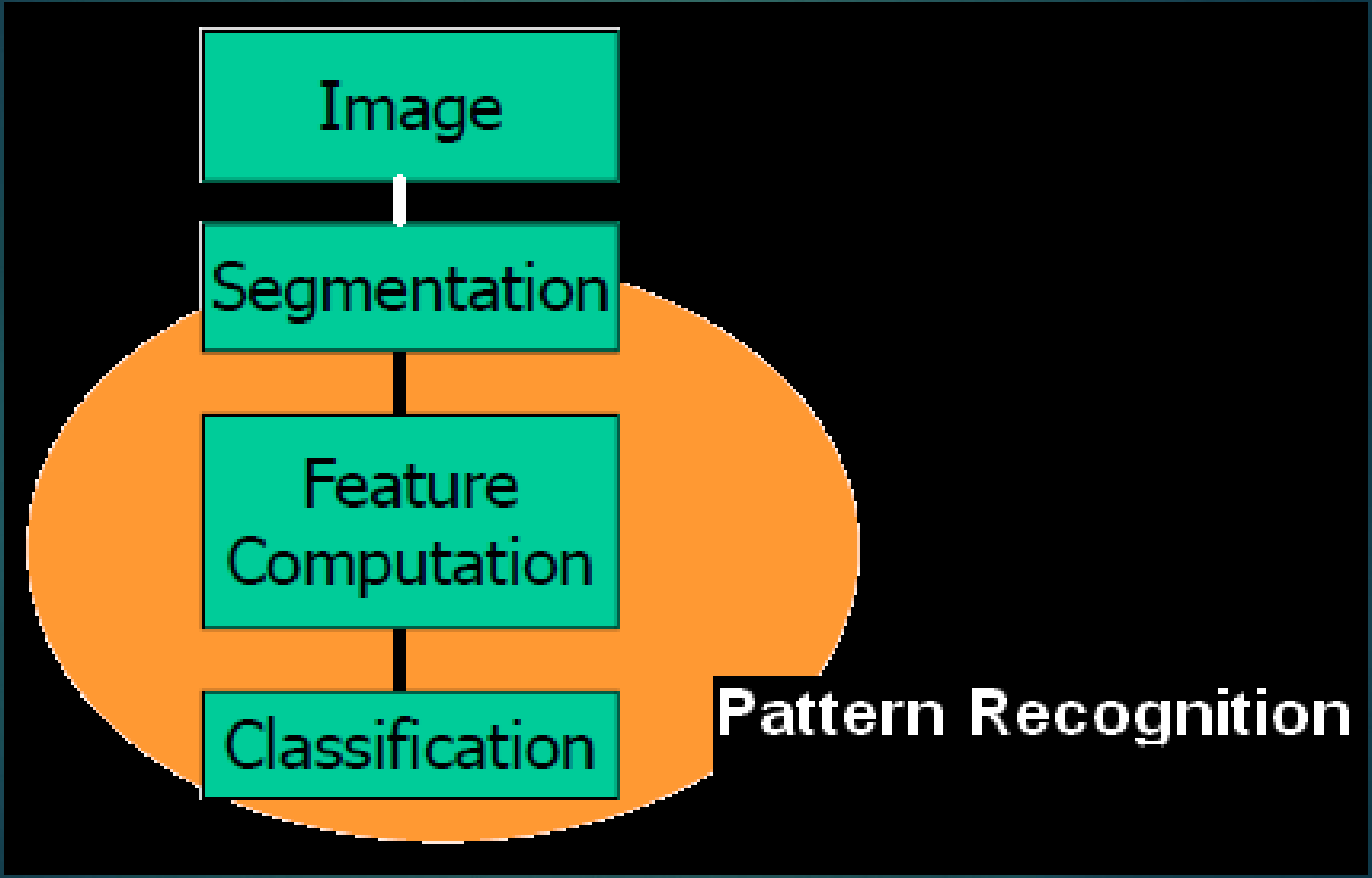
1. “A pattern is the opposite of chaos; it is an entity vaguely defined, that could be given a name.”

2. A pattern is an abstract object, such as a set of measurements describing a physical object.

3. ***Two key approaches:***

- Classification - we already know the classes
- Clustering - we don’t know the kinds of classes

1-Introduction



- **Classification**

- **Definition**: Classification is a **supervised learning** technique that involves categorizing data into predefined classes or labels.
- **Goal**: The goal of classification is to assign a label or category to each data point based on historical labeled data.
- **Examples**:
  - Email spam detection (classifying emails as "spam" or "not spam").
  - Image recognition (classifying images as "cat," "dog," etc.).
  - Medical diagnosis (classifying patients into "high risk" or "low risk" categories).

Common Algorithms: Decision trees, Support Vector Machines (SVM), K-nearest neighbors (KNN), and neural networks.

Clustering

- **Definition**: Clustering is an **unsupervised learning** technique that involves grouping data points into clusters based on similarity, without predefined labels.
- **Goal**: The goal of clustering is to organize data into natural groups or patterns, revealing hidden structures in the data.
- **Examples**:
  - Customer segmentation (grouping customers by purchasing behavior).
  - Image segmentation (dividing images into clusters of similar pixels).
  - Document clustering (organizing documents into topics).
- **Common Algorithms**: K-means, hierarchical clustering, DBSCAN, and Gaussian Mixture Models (GMM).

Image segmentation

- **Image segmentation** is a computer vision technique used to partition an image into meaningful segments, or regions, by grouping together pixels that share similar characteristics.
- Unlike general image classification, which assigns a single label to an entire image, segmentation provides a more detailed and localized understanding by labeling each pixel according to the object or region it belongs to within the image. This technique is crucial in applications requiring precise understanding and analysis of image content.

Type of segmentation

1. **Semantic Segmentation**:
   1. In semantic segmentation, each pixel is assigned to a specific class based on its semantic meaning. For example, in an image containing a dog and a cat, all pixels associated with the dog would be labeled "dog," and all pixels with the cat would be labeled "cat."
   2. **Applications**: Self-driving car perception (distinguishing road, pedestrians, vehicles), medical imaging (organ identification), satellite imagery (distinguishing water, land, vegetation).

- **Instance Segmentation**:
- Instance segmentation extends semantic segmentation by not only labeling each pixel by its class but also differentiating between different instances of the same class. For example, if there are multiple cats in an image, each cat would be labeled separately.
- **Applications**: Object detection for robotics, wildlife monitoring, and advanced scene understanding in autonomous vehicles.

- **Common Image Segmentation Algorithms**

1. **Thresholding**:
    1. Based on pixel intensity values, thresholding segments an image by converting it to a binary format. It works well for simple images with high contrast.
2. **Edge Detection-Based Segmentation**:
    1. Detects object boundaries in an image by identifying sharp changes in pixel intensity.
    2. Techniques: Canny edge detector, Sobel operator.
3. **Region-Based Segmentation**:
    1. Divides an image into regions based on criteria like intensity, texture, or color similarity.
    2. Techniques: Watershed algorithm, Region growing.
4. **Clustering-Based Segmentation**:
    1. Groups similar pixels together using clustering techniques like K-means.
5. **Deep Learning-Based Segmentation**:
    1. Uses convolutional neural networks (CNNs) to achieve high-accuracy segmentation.
    2. Architectures: Fully Convolutional Networks (FCNs), U-Net (commonly used in medical imaging), Mask R-CNN (for instance segmentation).

Feature extraction

- **Image feature extraction** is a process in computer vision where distinctive, informative features or characteristics of an image are identified and extracted.
- These features encapsulate important information about objects, textures, shapes, colors, and other patterns in the image, allowing for more efficient analysis, classification, or object recognition. Rather than working directly with raw pixel data, feature extraction reduces the image data to meaningful, compact representations, simplifying subsequent tasks like classification, segmentation, and object detection.

Types of Image Features

Color Features:

1. Represents the color distribution or patterns in an image, often through color histograms, color moments, or color spaces like RGB, HSV, or LAB.
2. **Example**: Using a color histogram to describe the proportion of different colors in an image.
3. **Applications**: Content-based image retrieval, scene classification.

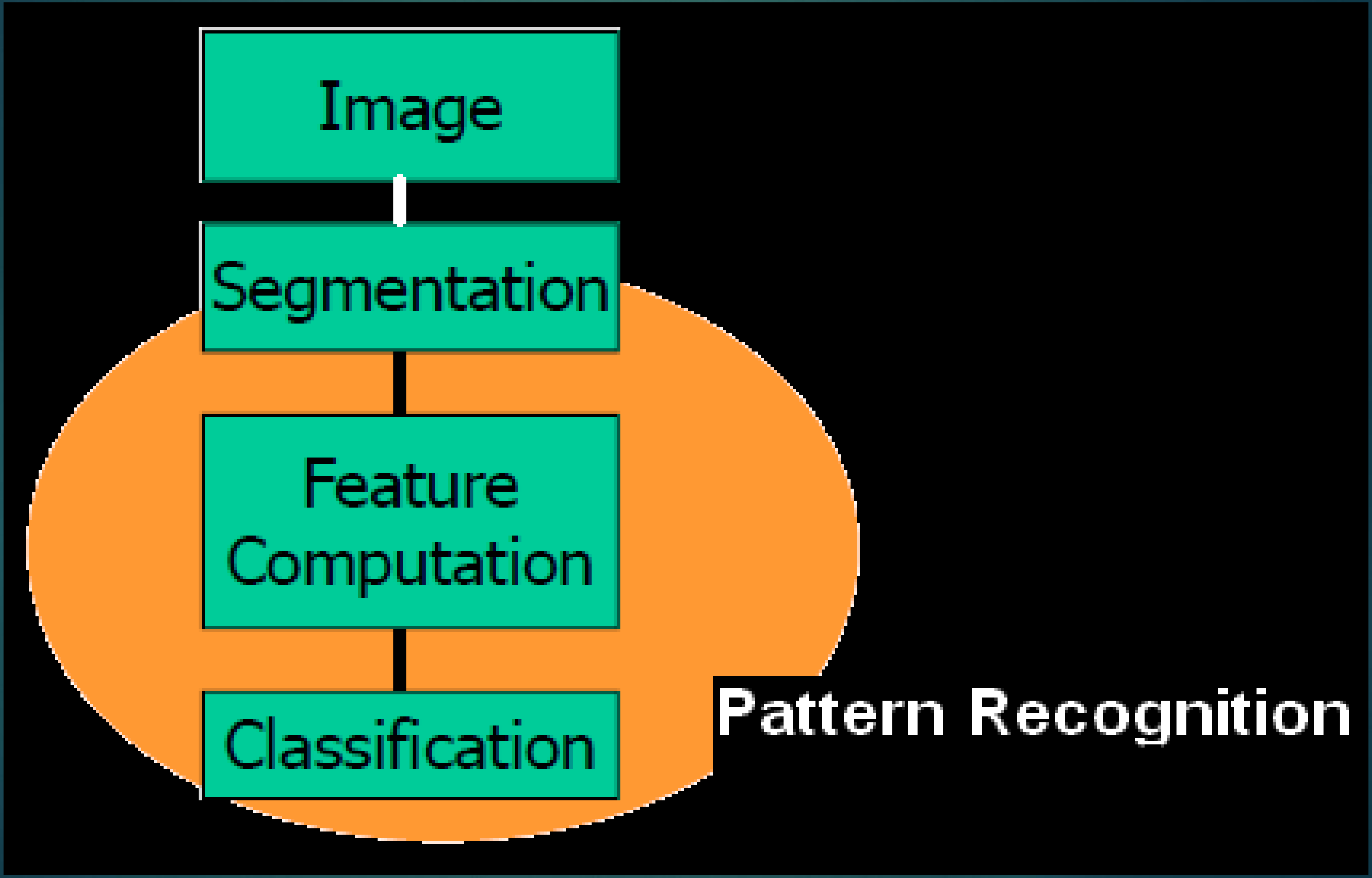
Texture Features:

- Describes patterns, such as roughness, smoothness, or regularity, in the image's surface.
- Techniques:
  - **Gray Level Co-occurrence Matrix (GLCM)**: Measures texture by analyzing how often pairs of pixel intensities occur.
  - **Local Binary Patterns (LBP)**: Captures texture by comparing each pixel with its neighbors.
- **Applications**: Medical imaging, face recognition, image classification.

Shape Features:

- Encodes the shape or contour of objects within an image.
- Techniques:
  - **Edge Detection**: Uses algorithms like the Canny or Sobel operators to detect boundaries between objects.
  - **Hough Transform**: Detects specific shapes like lines, circles, and ellipses.
  - **Contours and Blobs**: Used to find continuous curves or connected regions in an image.
- **Applications**: Object recognition, image segmentation, industrial quality control.

1-Introduction



1-Introduction

What is the Pattern Recognition?

- Pattern recognition is a field within the area of machine learning, and It **aims to classify** data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. Most of the pattern recognition methods that exist make use of procedures and algorithms as we will explain in detail later.
- The patterns to be classified are usually **groups** of **measurements** or **observations**, defining points in an appropriate multidimensional space.
- A **complete** pattern recognition system makes use of all the characteristics mentioned earlier: a **sensor** that gathers the observations to be classified or described; a feature extraction mechanism that computes numeric or symbolic information from the observations; and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features.

What is Pattern Recognition?

- Pattern recognition (PR) is the scientific system that concerns the description and classification (recognition) of patterns (objects)

- PR techniques are an important component of intelligent systems and are used for many application domains

  - Decision making

  - Object and pattern classification

Human Perception

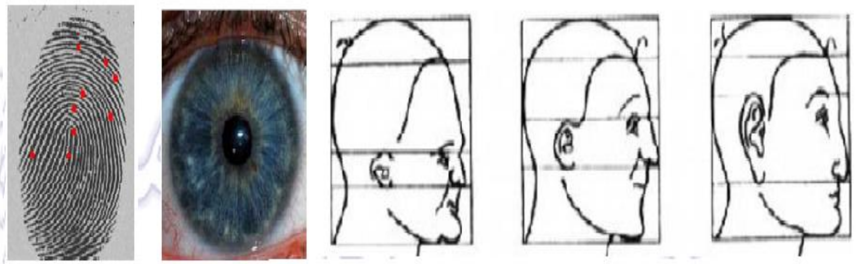
- Humans have developed highly sophisticated skills for sensing their environment and taking actions according to what they observe, e.g., recognizing a face, understanding spoken words, reading handwriting, distinguishing fresh food from its smell. We would like to give similar capabilities to machines

- Every day, we recognize faces around us, but we do it unconsciously and because we cannot explain our expertise, we find it difficult to write a computer program to do the same.

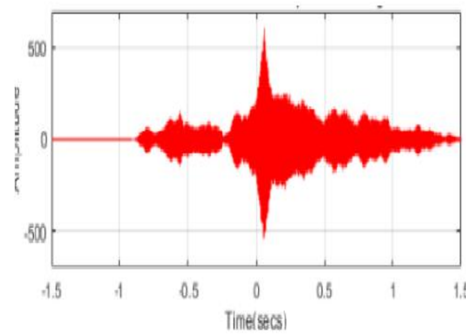
- Each person’s face is a **pattern** composed of a particular combination of structures (eyes, nose, mouth, . . .) located in certain positions on the face.

By analyzing sample images of faces, a program should be able to capture the pattern specific to a face and identify (or recognize) it as a face (as a member of a category or class we already know); this would be **pattern recognition**

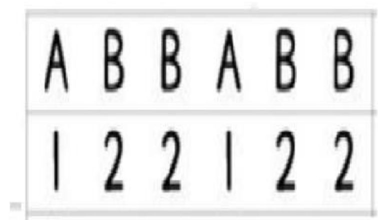
pattern (النموذج او النمط)



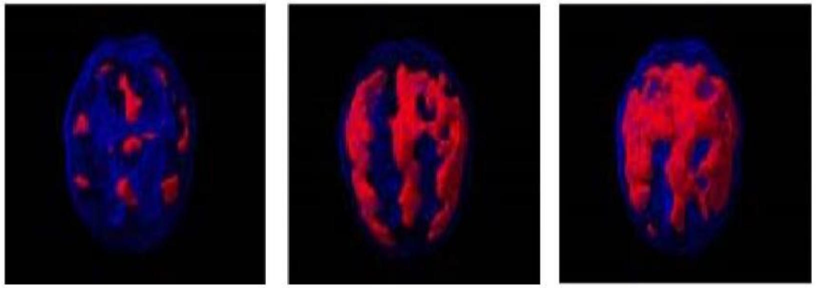
Biometric patterns (face, iris, fingerprint)



Signal patterns (sound pattern)



Character and number patterns



Medical patterns

For example, an MRI of the brain includes expressive patterns of brain activity that, by identifying and analyzing them, can predict brain behaviors and activities and estimate disease incidence.

A **class** is a collection of objects that are similar, but not necessarily identical, and which is distinguishable from other classes. Fig.1 illustrates the difference

between classification where the classes are known beforehand and classification where classes are created after inspecting the objects

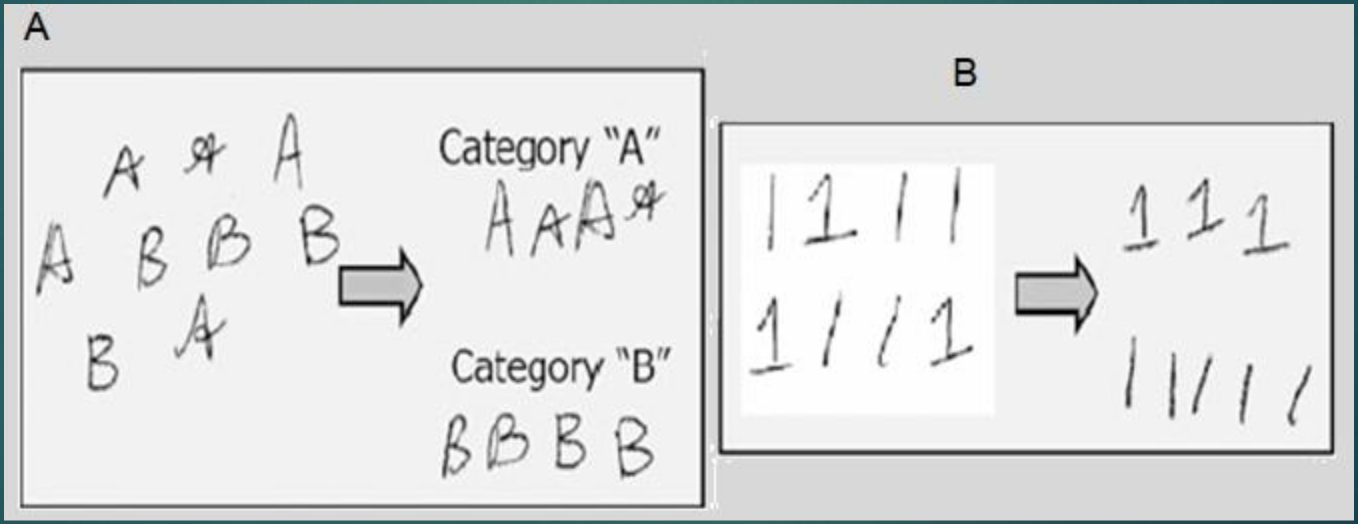


Fig 1 Classification when the classes are (A) known and (B) unknown beforehand

Human and Machine Perception

- We are often affected by the knowledge of **how patterns are modeled and recognized in nature** when we develop pattern recognition algorithms. Research on machine perception also helps us gain a deeper understanding and appreciation for pattern recognition systems in nature. Yet, we also apply many techniques that are purely numerical and do not have any correspondence with natural systems.

Machine Perception

- Build a machine that can recognize patterns:
  - Speech recognition
  - Fingerprint identification
  - OCR (Optical Character Recognition)
  - DNA sequence identification

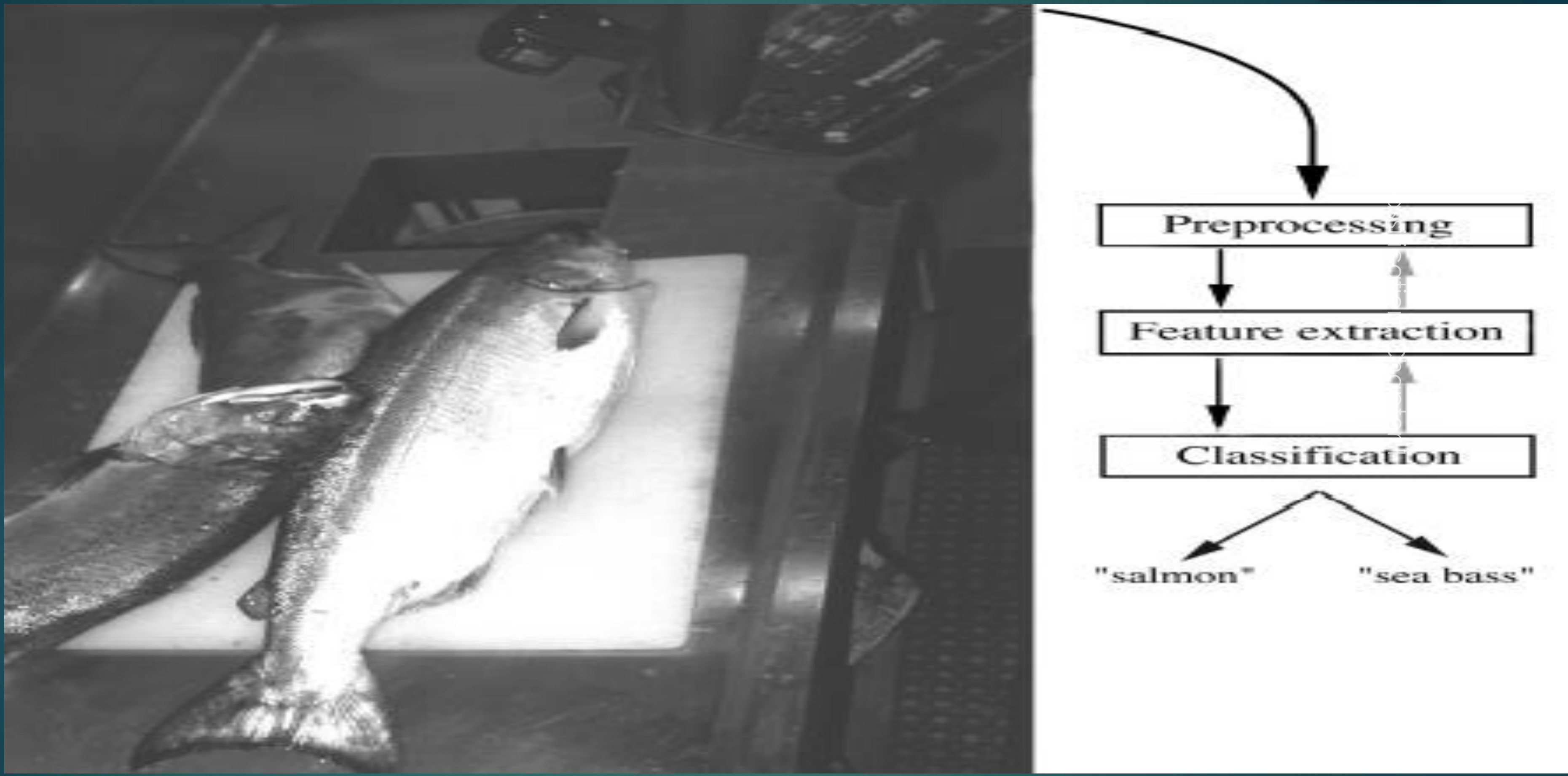
An Example

- "Sorting incoming Fish on a conveyor according to species using optical sensing"

Sea bass

Species

Salmon


Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

- **Preprocessing**

  - Use a **segmentation** operation to isolate fishes from one another and from the background

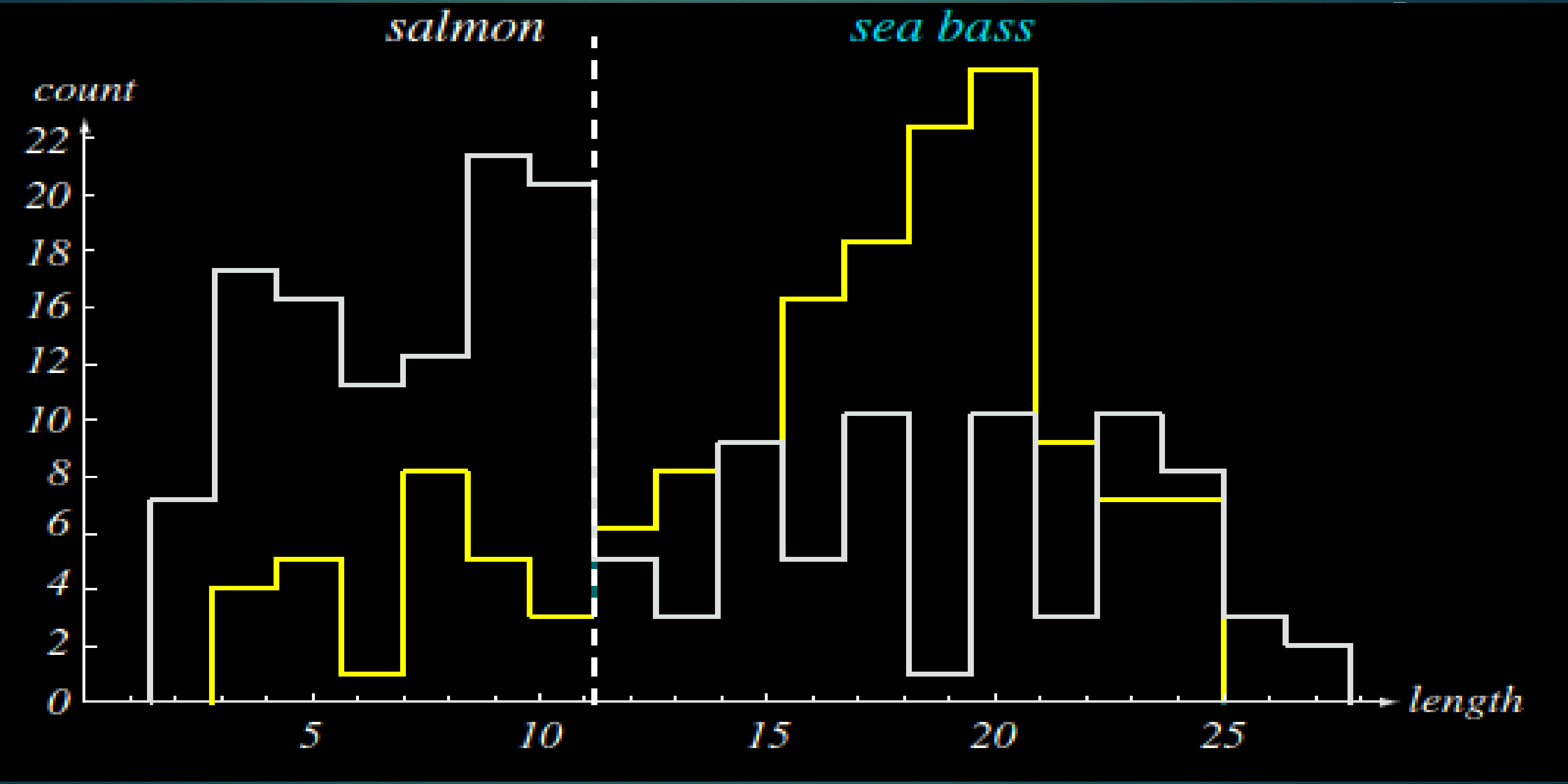
- Information from a single fish is sent to a **feature extractor** whose purpose is to reduce the data by measuring certain features

- The features are **passed** to a **classifier**

- Problem Analysis
  - Set up a camera and take some sample images to extract features
    - **Length**
    - **Lightness**
    - **Width**
    - **Number and shape of fins**
    - **Position of the mouth, etc…**
    - This is the set of all suggested features to explore for use in our classifier!

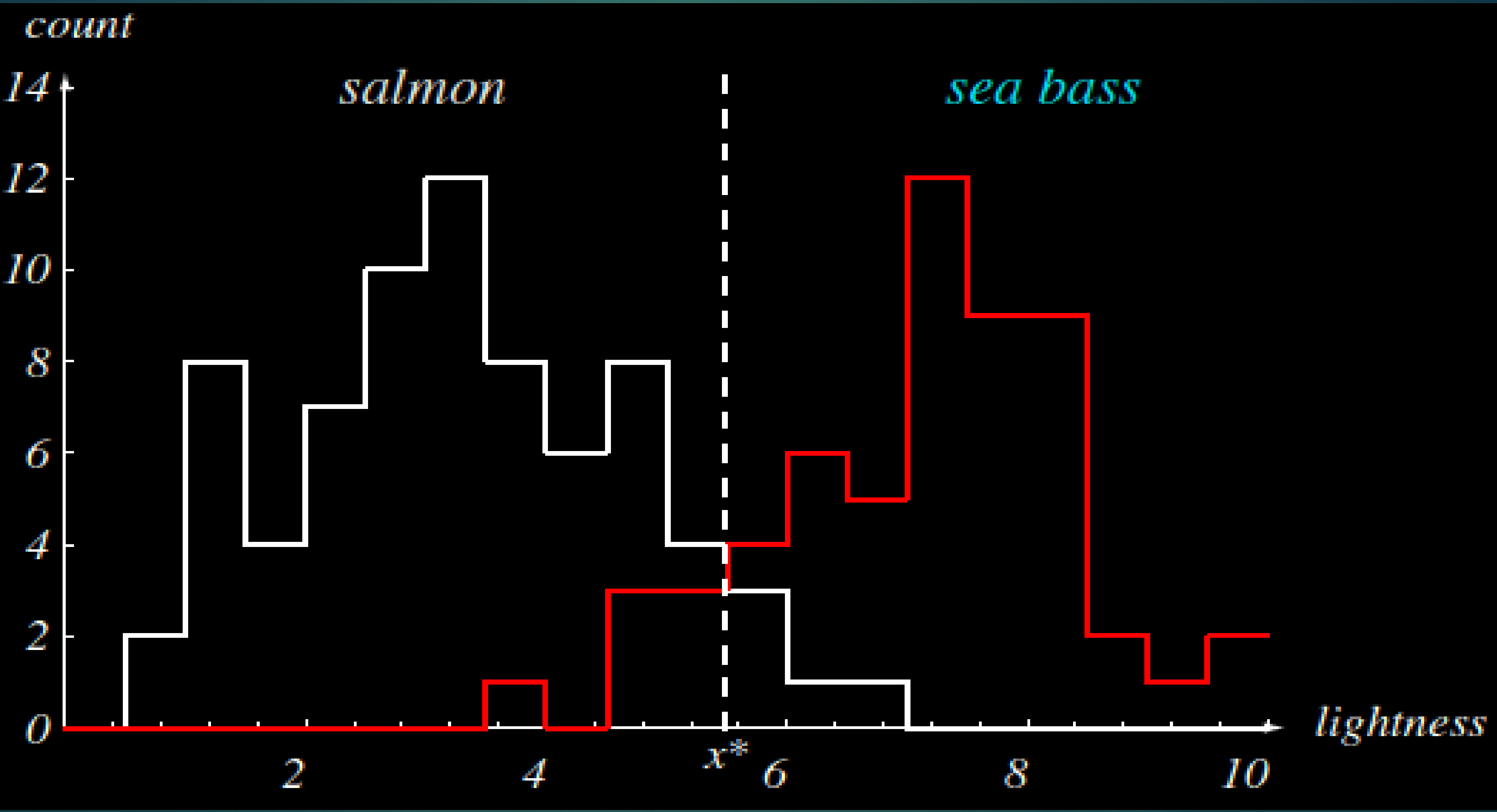
Classification

- Select the length of the fish as a possible feature for discrimination



The **lightness** is a poor feature alone!

Select the **lightness** as a possible feature.



- Threshold decision boundary and cost relationship
  - Move our decision boundary toward smaller values of lightness in order to minimize the cost (reduce the number of sea bass that are classified salmon!)

Task of decision theory

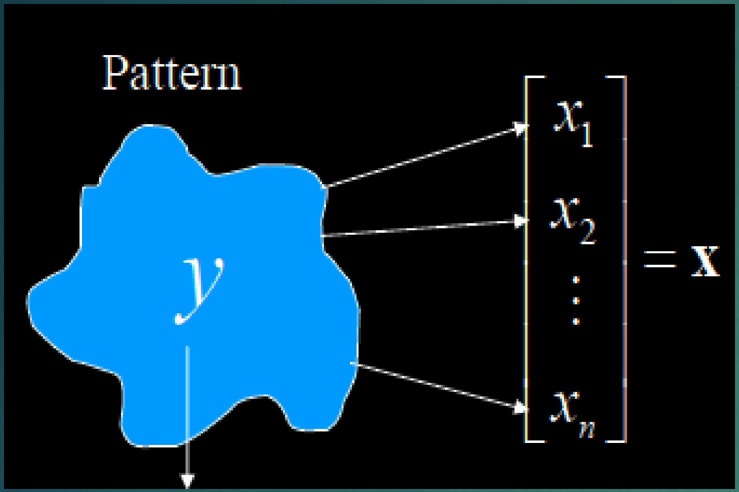
- Feature extraction

Task: to extract features which are good for classification.

Good features:

- Objects from the same class have similar feature values.
- O es.

Therefore… Basic concepts



Feature vector : $\mathbf{x} \in \mathbf{X}$

- A vector of observations (measurements)
- $\mathbf{x}$ is a point in feature space $\mathbf{X}$

Hidden state: $\mathbf{y} \in \mathbf{Y}$

- Cannot be directly measured.

-Patterns with equal hidden states belong to the same class.

Task

- To design a classifier (decision rule) $\mathbf{q : X \rightarrow Y}$

which decides about a hidden state based on an observation.

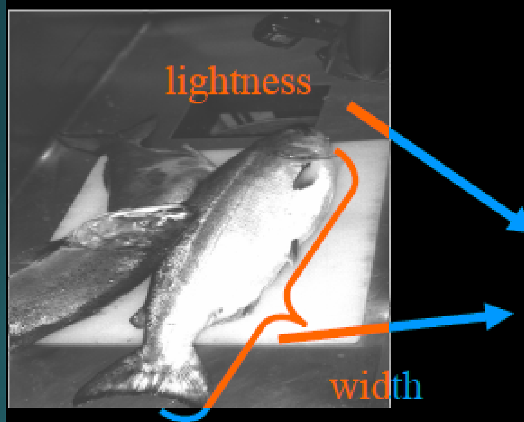
- Adopt the lightness and add the **width** of the fish

Fish $\Rightarrow$ $x^T = [x_1, x_2]$

Lightness **Width**

An example of Industrial Inspection

In our case…



$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{x}$$

Sea bass

Salmon

Task: fish recognition.

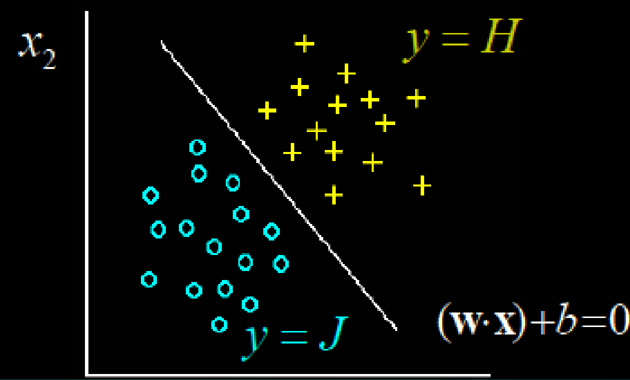
The set of hidden state is $Y = \{H, J\}$

The feature space is $X = \Re^2$

Training examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$

Linear classifier:

$$q(\mathbf{x}) = \begin{cases} H & if \quad (\mathbf{w} \cdot \mathbf{x}) + b \geq 0 \\ J & if \quad (\mathbf{w} \cdot \mathbf{x}) + b < 0 \end{cases}$$



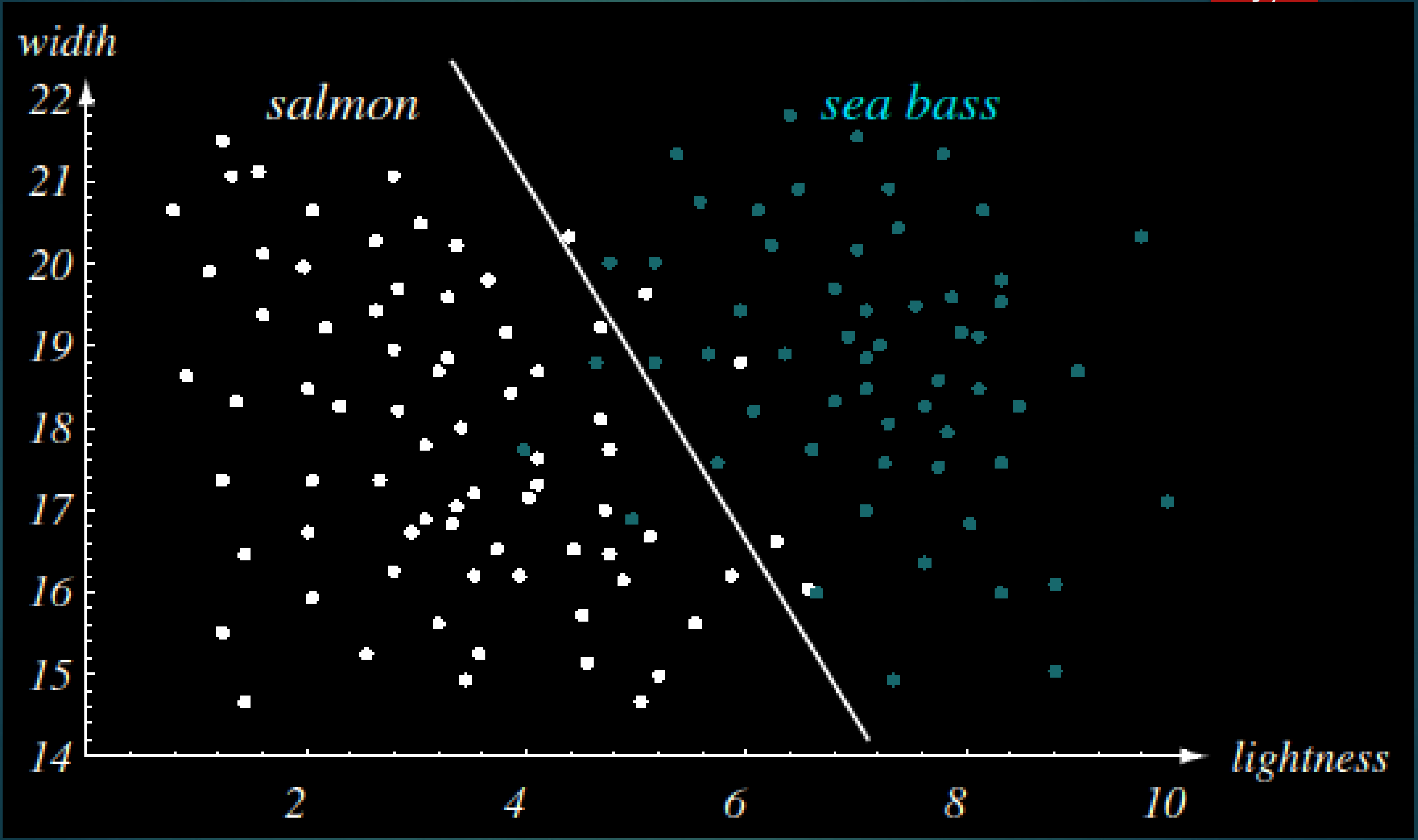
Simple Salmon vs. Sea Bass Classifier

- Consider a simple classifier that uses only two features, such as **average color intensity** and **length**:
- **Feature Extraction**:
  - Measure the average color intensity of each fish image and record the length.
- **Decision Rule**:
  - Suppose salmon are generally lighter and shorter, and sea bass are darker and longer.
  - A decision rule could be set as:
    if (color intensity > threshold) and (length < threshold), classify as S almon; else
  - classify as Sea Bass\text{if (color intensity > threshold) and (length < threshold), classify as Salmon; else classify as Sea Bass}if (color intensity > threshold) and (length < threshold), classify as Salmon;
  - else classify as Sea Bass

Threshold in Pattern Recognition (Salmon vs. Sea Bass)

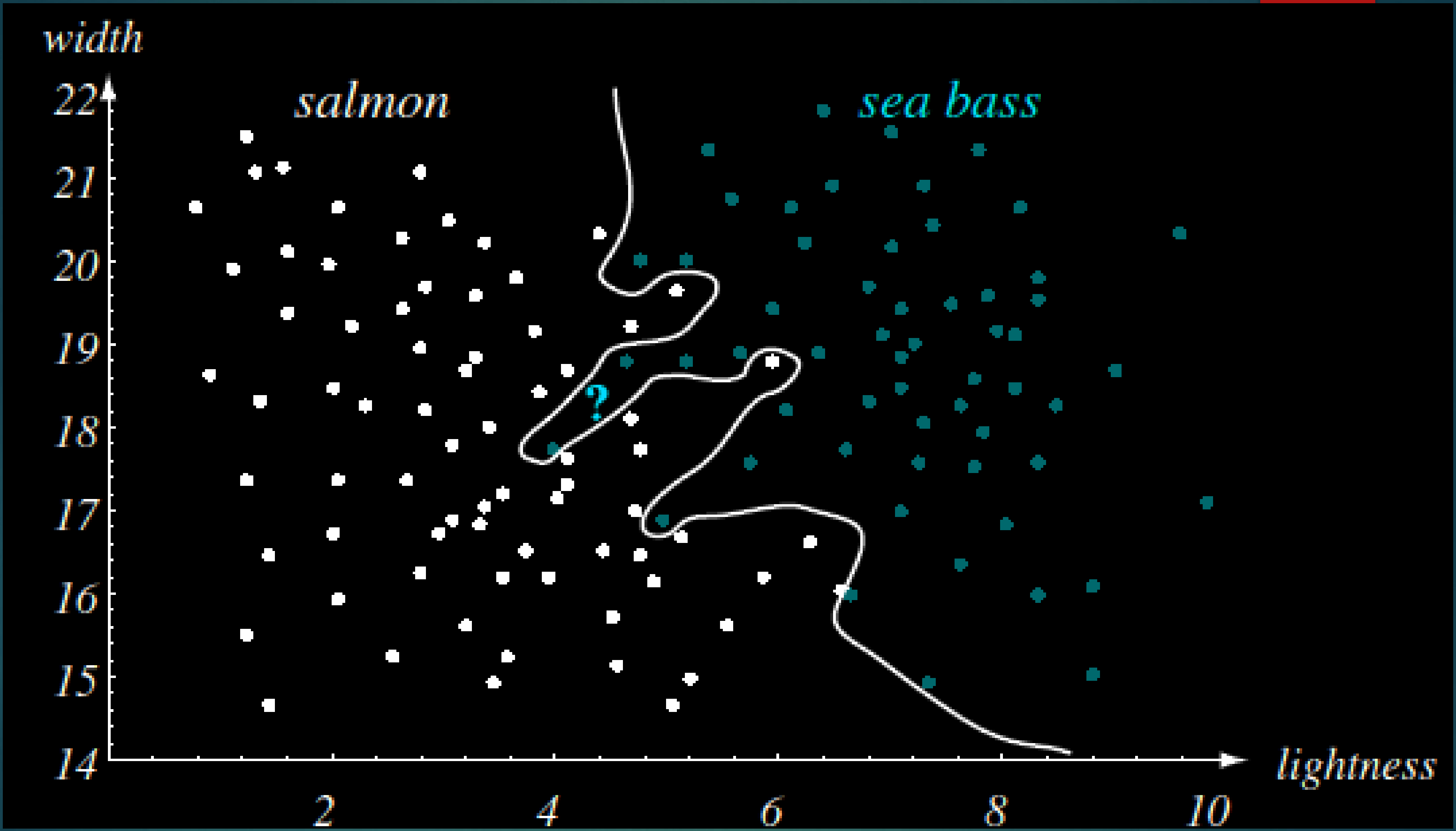
- For the **salmon vs. sea bass** example, let's say we have a feature called **color intensity** (how light or dark the fish appears on average). Based on observations or data analysis, we might notice that salmon tend to have a color intensity above a certain value (they are generally lighter), while sea bass tend to fall below that value.
- **Threshold Selection**: Suppose we choose a threshold value of **0.5** for color intensity (on a scale from 0 to 1).
- **Decision Rule**:
  - If the **color intensity** of a fish is **greater than 0.5**, classify it as **salmon**.
  - If the **color intensity** is **less than or equal to 0.5**, classify it as **sea bass**.

Linear (simple) decision boundary; Cost of misclassification



- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such “noisy features”

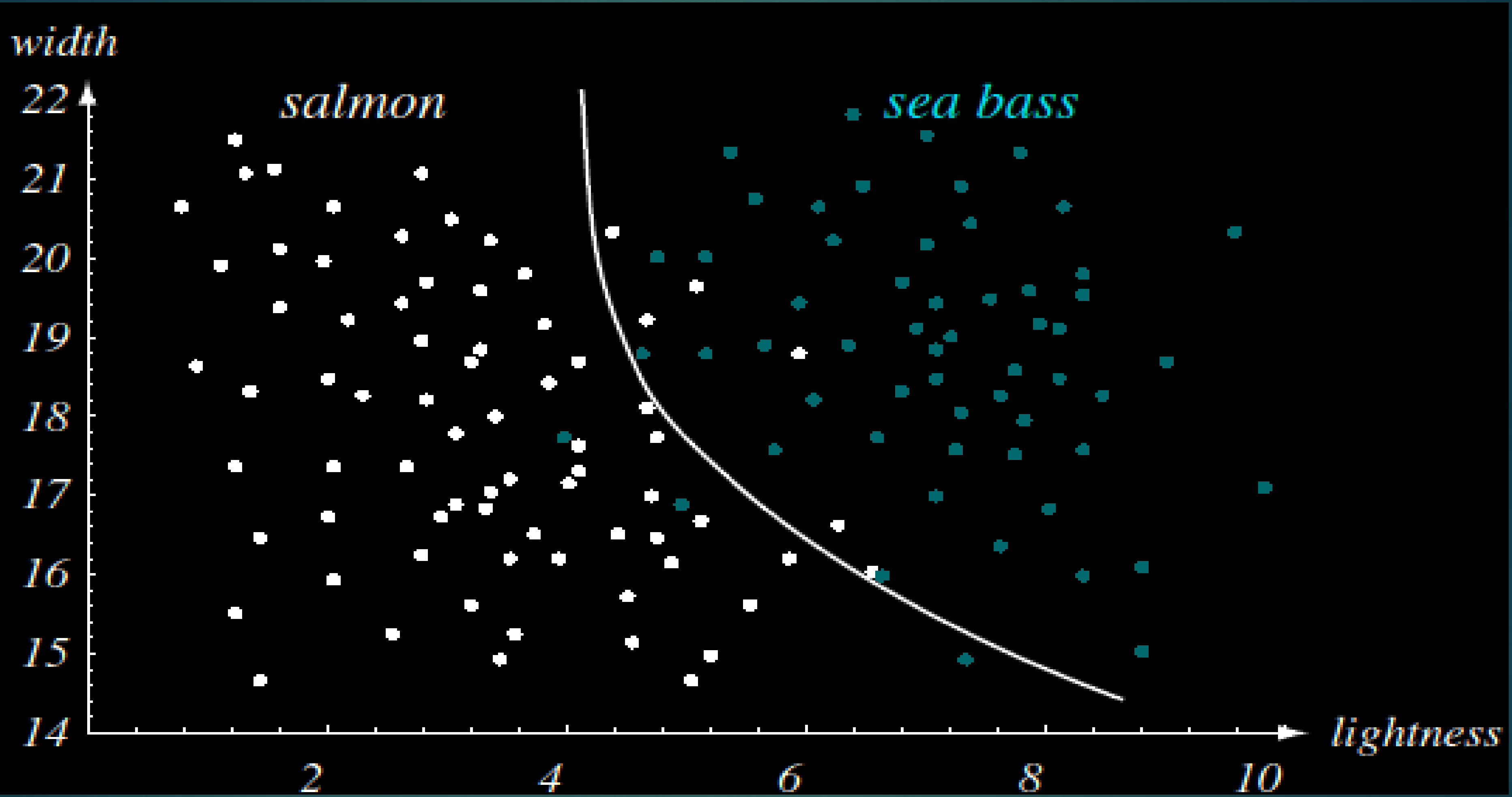
- Ideally, the best decision boundary should be the one that provides an optimal performance such as in the following figure:


width
salmon
sea bass
?
lightness
22
21
20
19
18
17
16
15
14
2
4
6
8
10

- However, our **satisfaction** is **premature** because the central aim of designing a classifier is to correctly classify novel input

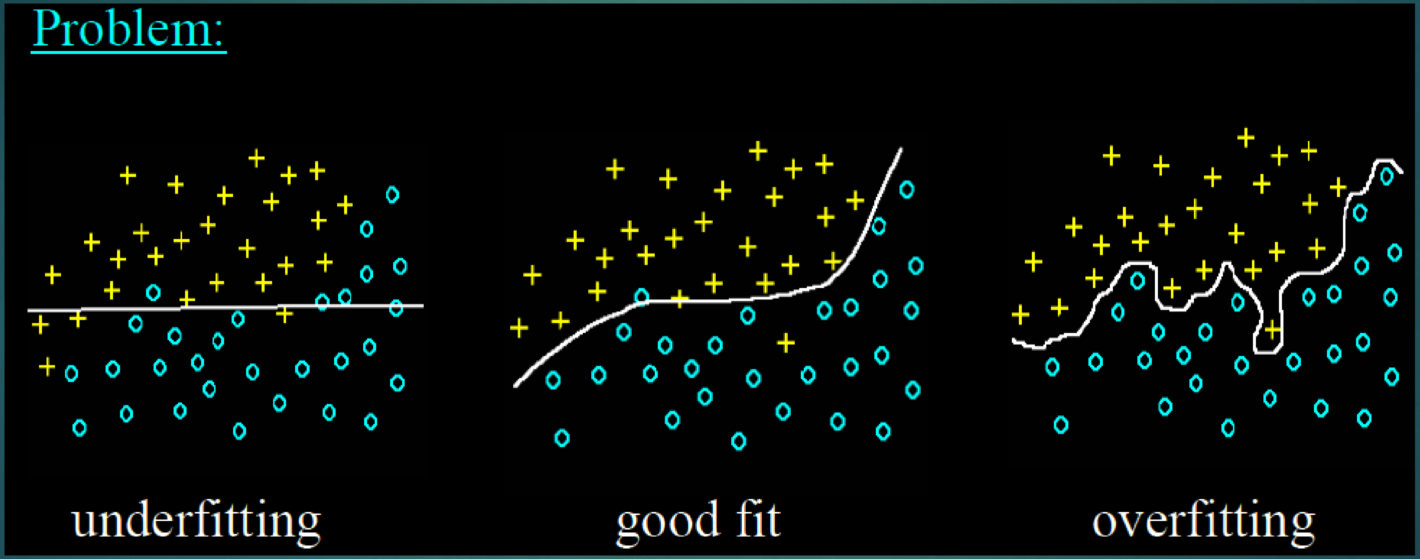
Issue of **generalization!**

An example of Industrial Inspection



An example of Industrial Inspection

Overfitting and underfitting



Overfitting

Overfitting terjadi ketika model belajar terlalu banyak dari data pelatihan, bahkan menangkap detail atau “noise” yang tidak relevan. Model jadi terlalu kompleks dan terlalu "spesifik" pada data pelatihan sehingga kurang mampu bekerja baik pada data baru.

- **Ciri-ciri**:
  - Akurasi tinggi pada data pelatihan tetapi rendah pada data pengujian.
  - Biasanya terjadi pada model dengan banyak parameter atau model yang terlalu kompleks.
- **Solusi**:
  - Gunakan regularisasi (misalnya, L1 atau L2 regularization).
  - Sederhanakan model atau kurangi jumlah fitur.
  - Perbanyak data pelatihan atau gunakan teknik cross-validation.

Underfitting

- Underfitting terjadi ketika model terlalu sederhana sehingga tidak dapat menangkap pola dalam data pelatihan dengan baik. Model ini biasanya kurang akurat baik pada data pelatihan maupun data pengujian.
- **Ciri-ciri**:
  - Akurasi rendah pada data pelatihan dan data pengujian.
  - Model dengan sedikit parameter atau terlalu sederhana sering mengalami underfitting.
- **Solusi**:
  - Tingkatkan kompleksitas model (misalnya, gunakan model yang lebih kompleks atau tambahkan parameter).
  - Tingkatkan durasi pelatihan atau kurangi regularisasi jika terlalu kuat

Emerging Applications

Interest in pattern recognition and classification has grown due to emerging applications, which include:-

- Data mining (sifting through a large volume of data to extract a small amount of relevant and useful information, e.g., fraud detection, and financial forecasting).
- Biometrics (personal identification based on physical attributes of the face, iris, fingerprints, etc.)
- Machine vision (e.g., automated visual inspection in an assembly line)
- Character recognition [(automated teller machines)]
- Document recognition (e.g., recognize whether an e-mail is a spam or not, based on the message header and content)

Good Fitting

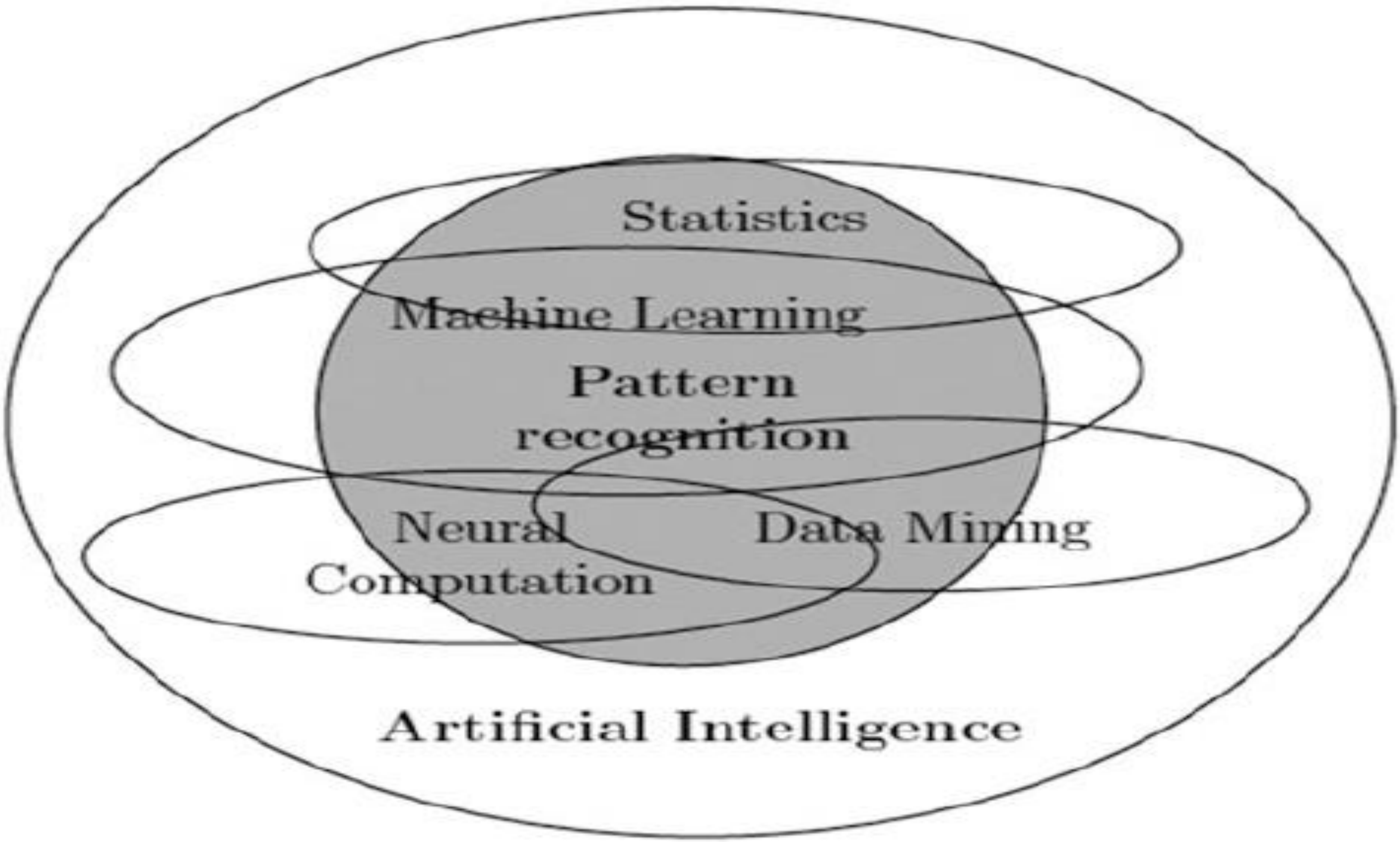
- Good fitting adalah kondisi di mana model mampu mengenali pola utama dalam data pelatihan tanpa menangkap noise yang tidak relevan, sehingga hasilnya baik pada data pelatihan dan data pengujian.
- **Ciri-ciri**:
  - Akurasi pada data pelatihan dan data pengujian hampir sama dan cukup tinggi.
  - Model bisa memprediksi data baru dengan baik tanpa menjadi terlalu spesifik atau terlalu umum.

- Computer-aided diagnosis [e.g., helping doctors make diagnostic decisions based on interpreting medical data such as mammographic images.
- Medical imaging [e.g., classifying cells as malignant or benign based on the results of magnetic resonance imaging (MRI) scans.
- Speech recognition (e.g., helping handicapped patients to control machines).
- Bioinformatics (e.g., DNA sequence analysis to detect genes related to particular diseases).
- Remote sensing (e.g., land use and crop yield).
- Astronomy (classifying galaxies based on their shapes

The methods used have been developed in various fields, often independently.

- In statistics, going from particular observations to general descriptions is called **inference, learning** [i.e., using example (training) data] is called **estimation**, and **classification** is known as **discriminant analysis**.
- In engineering, classification is called **pattern recognition** and the approach is nonparametric and much more empirical. Other approaches have their origins in machine learning, artificial intelligence, artificial neural networks, and data mining. Fig. 2 illustrated the Pattern recognition and related fields

Fig. 2 Pattern recognition and related fields



Classification

- Classification is often the final step in a general process (Fig. 3). It involves sorting objects into separate classes.

In the case of an image, the acquired image is segmented to isolate different objects from each other and from the background, and the different objects are labeled.

The goal of the classifier is to classify new data (test data) to one of the classes, characterized by a decision region. The borders between decision regions are called decision boundaries.

Classification techniques can be divided into two broad areas: statistical and structural (or syntactic) techniques, with a third area that borrows from both , sometimes called cognitive methods, which include neural networks and genetic algorithms

- A typical pattern recognition system contains a **sensor, a pre-processing mechanism (prior to segmentation), a feature extraction mechanism, a set of examples (training data) already classified (post-processing), and a classification algorithm**. As shown in Fig.3

Pattern Recognition Systems

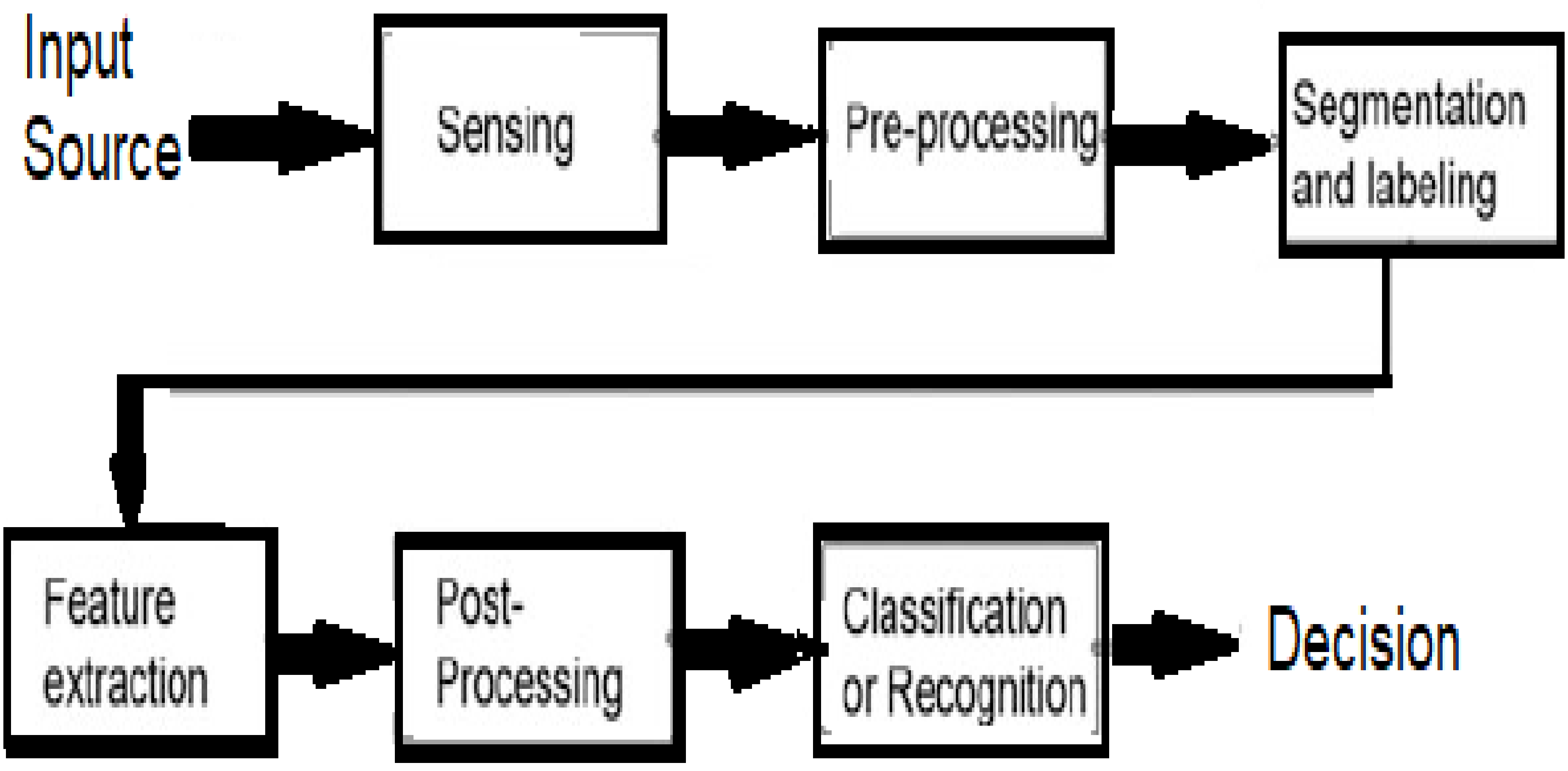


Fig.3:- A general classification system

Pattern Recognition Systems

- **Data acquisition and sensing**:

  - Use of a transducer (camera or microphone)

  - Important issues: bandwidth, resolution, sensitivity, distortion, SNR, latency, etc.

- **Pre-processing:**

  -Removal of noise in data.

  - Isolation of patterns of interest from the background

- Segmentation and grouping

  Patterns should be well separated and should not overlap

Pattern Recognition Systems

- **Feature extraction**

  Finding a new representation in terms of features

  - Discriminative features

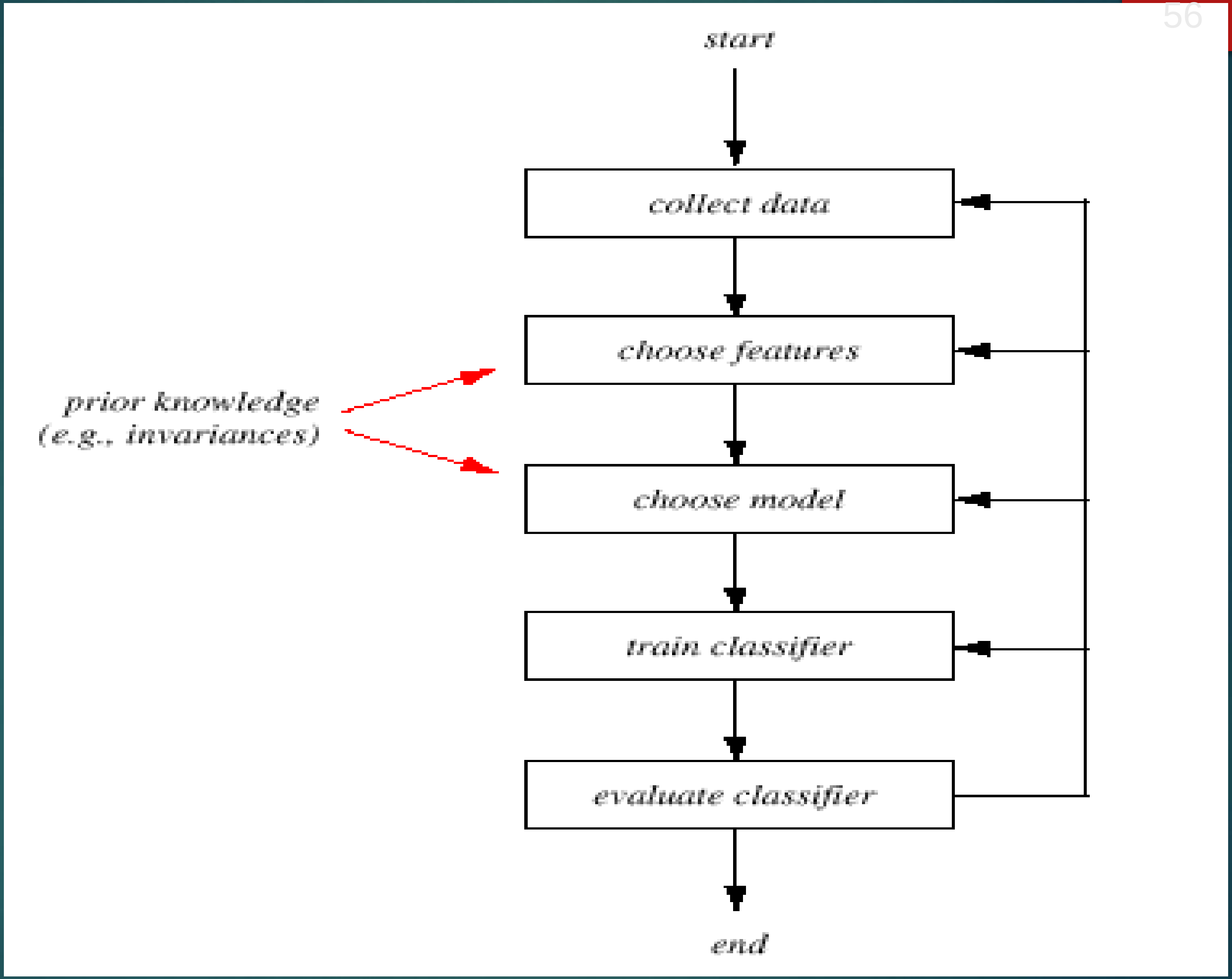
  - Invariant features with respect to translation, rotation and scale.

- **Post Processing**
  - Exploit context input dependent information other than from the target pattern itself to improve performance

- **Classification**
  - Use a feature vector provided by a feature extractor to assign the object to a category

The Design Cycle

- Data collection
- Feature Choice
- Model Choice
- Training
- Evaluation
- Computational Complexity


start
collect data
choose features
prior knowledge
(e.g., invariances)
choose model
train classifier
evaluate classifier
end

- **Data Collection**

  - Collect an adequately large and representative set of examples and divide them into (70%) training and (30%) testing the system.

- **Feature Choice**

- Depends on the characteristics of the problem domain. *Simple* to extract, *invariant* to irrelevant transformation *insensitive* to noise

- **Model Choice**

Unsatisfied with the performance of one classifier, *jump* to *another* class of models

- **Training**

-Use *sample data* to train the classifier.

- Use *Many different* procedures for training classifiers and choosing models :

- Random Sub-sampling
- Bootstrap
- Cross-Validation

- **Evaluation**

– *Measure* the *error rate (performance*) using different training methods.

– *Switch* from one set of *features* to another, or from one *model* to another to *improve accuracy*, i.e. to *minimize error rate*.

Performance of PR Systems

- ***Error rate*** **(Prob. of *misclassification*)**
- ***Speed***
- ***Cost***
- ***Robustness***
- ***Reject option***
- ***Return on investment***

Computational Complexity

- What is the *trade-off* between *computational ease* and *performance?*

- How does an algorithm *scale* as a function of the number of *features, patterns* **or** *categories***?**

Limitation of PR Systems

- Humans have the *ability to switch rapidly* and *seamlessly* between different pattern recognition tasks.

- It is *very difficult* to design a device that is capable of performing a variety of different classification tasks as humans.

Learning and Adaptation

- Supervised learning
  - A teacher provides a category label or cost for each pattern in the training set
- Unsupervised learning
  - The system forms clusters or “natural groupings” of the input patterns

Summary

- Pattern recognition is *extremely useful* and are now *part* of many *crucial* computer applications:
- Pattern recognition is a *very difficult* problem and have many *complex sub-problems*.
- Successful systems have been built in *well constrained* domains.
- No *single technique/model* is suited for *all* pattern recognition problems
- Use of *object models*, *constraints*, and *context* is necessary for identifying *complex patterns*
- *Careful sensor design* and *feature extraction* can lead to *simple* classifiers