COMP 9517 Computer Vision

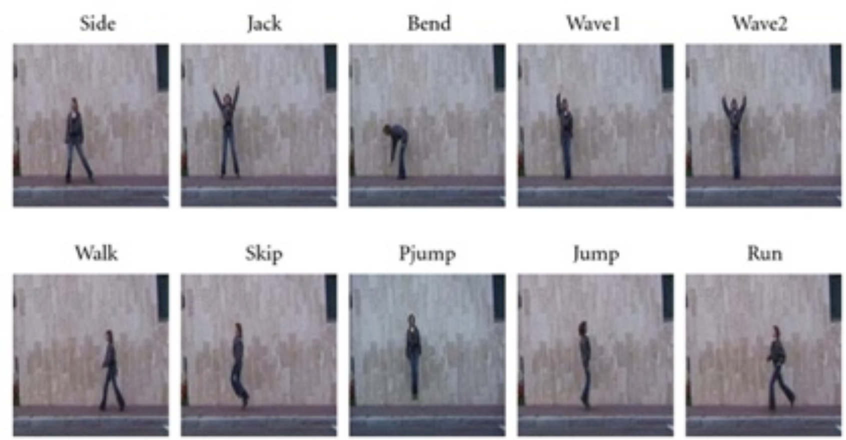
Pattern Recognition (1)

Introduction

- ***Pattern recognition*** is the scientific discipline whose goal is the classification of objects into a number of categories or classes
- Pattern recognition used widely for object classification and recognition
  - To recognise a face
  - To read handwritten characters
  - To identify our car keys in our pocket by feel
  - To understand spoken words
- Objects can be images or any type of measurements that need to be classified, which are referred using the generic term ***pattern***

Applications

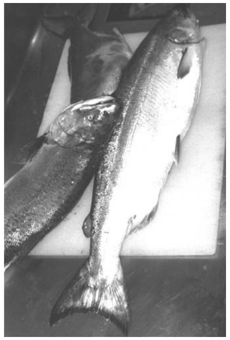
- Computer vision is an area in which pattern recognition is of importance
  - Making decisions about image content
  - Classifying objects in an image
  - Recognising activities

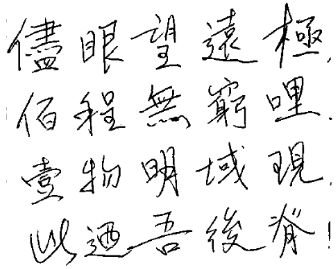


Images extracted from google.com

Applications

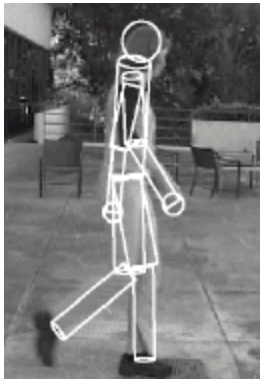
- Examples of pattern recognition in computer vision:
  - Machine vision
  - Character recognition
  - Face recognition
  - Human activity recognition
  - Image-based medical diagnosis





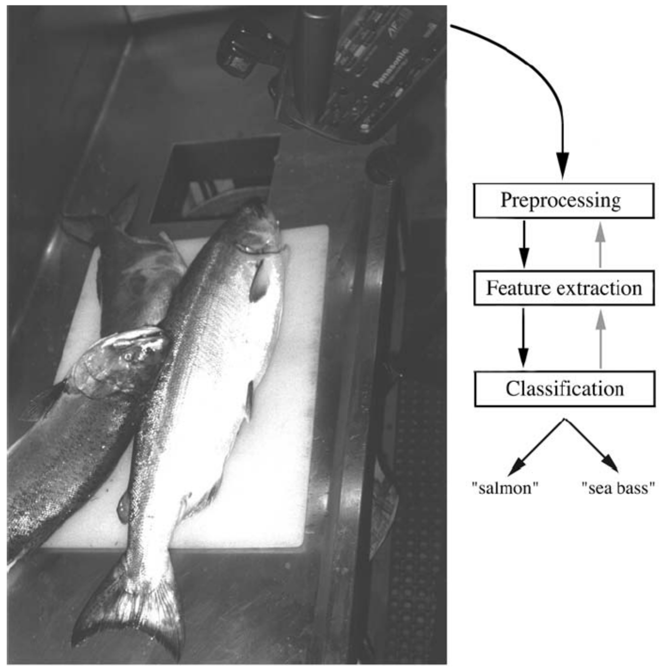
I looked as hard as I could see, beyond 100 plus infinity an object of bright intensity- it was the back of me!

- Other areas beside CV
  - For example, recommender systems

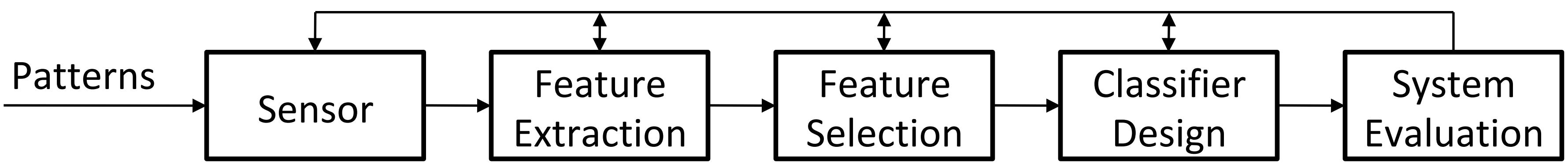


Pattern Recognition Systems

- Prototype of pattern recognition

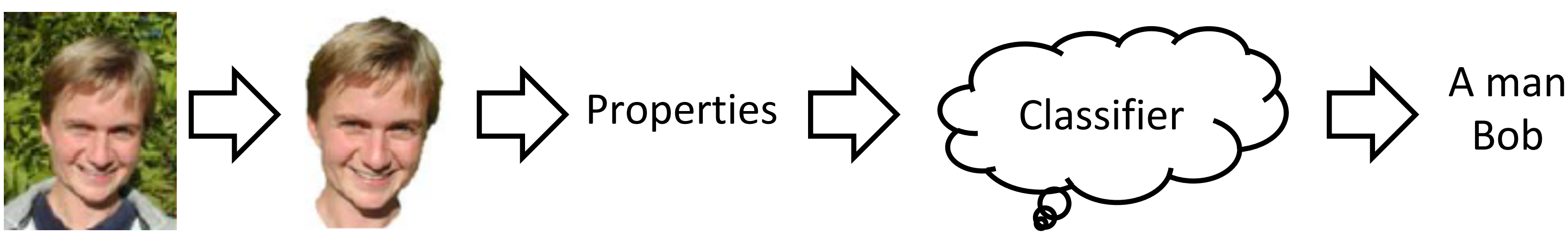


- The basic stages involved in the design of a classification system



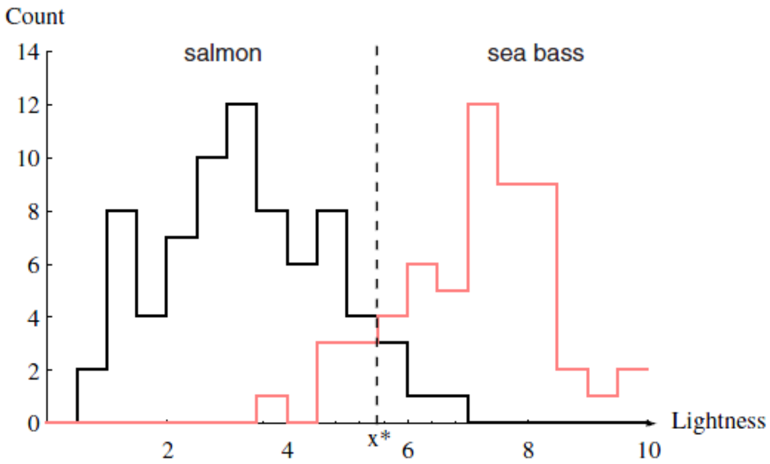
Pattern Recognition Concepts

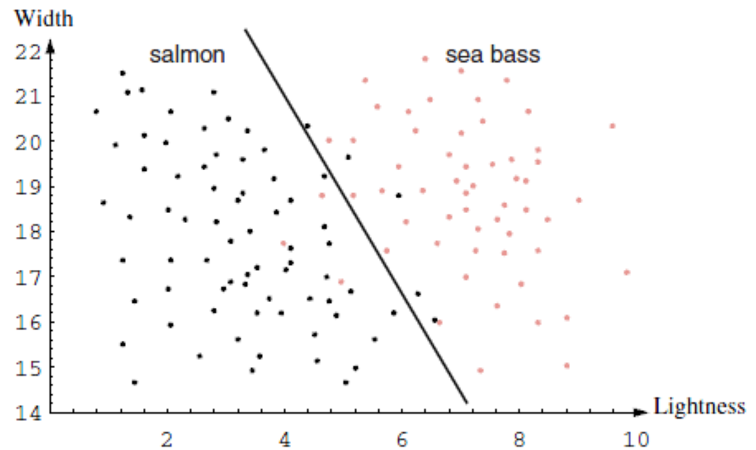
- ***Object*** - an object is a physical unit
- ***Regions*** - that correspond to objects are obtained, after segmentation of an image
- ***Classes*** - the set of objects can be divided into disjoint subsets that may have some common features- such sets are called classes
- ***Object recognition/pattern recognition*** - object recognition assigns classes to objects
- ***Classifier*** - the corresponding algorithm/method is called the classifier
- ***Pattern*** - the classifier bases its decision on object features, called the pattern



More Concepts

- ***Features*** - description of the objects
- ***Model*** - description of the classes
- ***Pre-processing*** - noise removal, segmentation
- ***Feature Extraction*** - reduce the data by measuring certain " features" or properties
- ***Training samples*** - experience, objects with known ground truth
- ***Cost*** - consequence of making incorrect decision
- ***Decision boundary*** - boundary between regions in feature space





Features and Descriptions

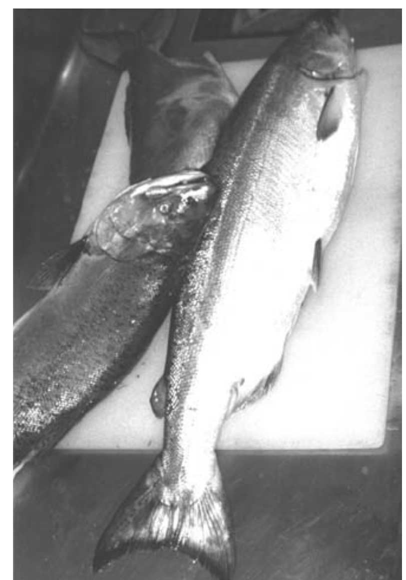
- ***Features***
  - descriptions representing scalar properties of objects are called ***features***
  - used to represent knowledge as part of more complex representation structure
- ***Feature vector***
  - combines many features, e.g. size feature represents area property, compactness feature represents circularity
- Good representation is important to solve a problem
- Rich structured representation can simplify control strategies

Feature Vector Representation

- $X=[x_1, x_2, \ldots, x_n]$, each x_j is a real number
  - x_j may be an object measurement
  - x_j may be count of object parts
- Example:
  - *[#holes, #strokes, moments, …]*
  - *[length, colour ,lightness,…]*

```
00000000010000000000
00000000110000000000
00000000101000000000
00000001000010000000
00000010000010000000
00000100000001000000
00001000000000100000
00001100111111110000
00001111111000010000
00011000000000011000
00010000000000001100
00110000000000000100
00110000000000000110
00100000000000000010
00100000000000000010
01100000000000000010
01000000000000000000
00000000000000000000
```

```
00000000001111000000
00000001100001100000
00000011000000110000
00001100000000011000
00001000000000001000
00001100000000010000
00000111000000100000
00000011100111100000
00000000111100000000
00000011000111000000
00001100000000110000
00011000000000011000
00110000000000001000
00100000000000001100
00010000000000011000
00011000000000010000
00001000000000110000
00000011111110000000
```



Feature Extraction

- Goal of feature extraction is to characterise object by measurements that are
  - similar for objects in the same class/category, and
  - different for objects in different classes
- Must find ***distinguishing features*** that are invariant to input transformations
- Design of features often based on prior experience or intuition

Feature Extraction

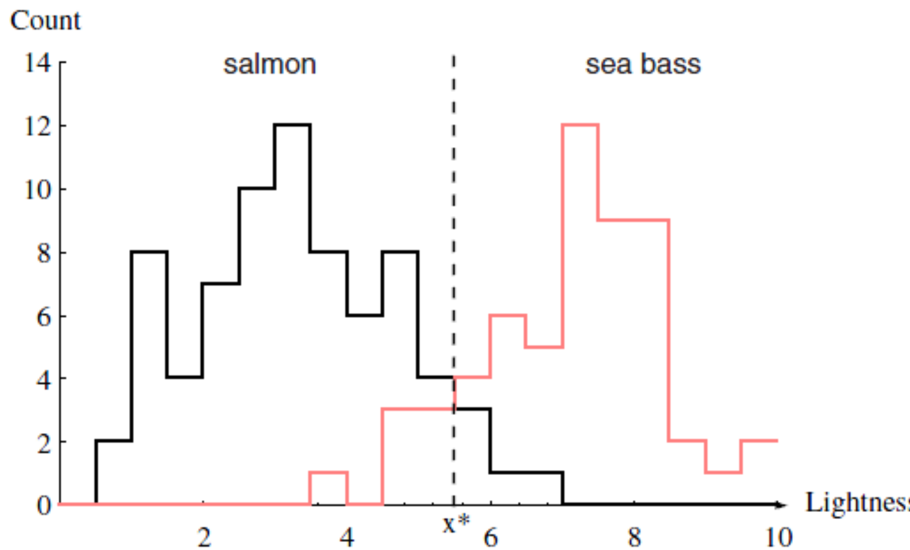
- Selecting features that are
  - translation, rotation and scale invariant in images
  - handling ***occlusion***, projective distortion for 3-D objects in images
  - invariant to translations in time and changes in amplitude
  - handling ***non-rigid deformations*** common in 3-D vision
- Feature selection is problem- and domain-dependent
- But classification techniques can help to
  - make feature values less noise sensitive, and
  - to select valuable features out of a larger set

Classification

- Classifier performs object recognition by assigning an object to a class
  - using the object description in the form of features
- Perfect classification is often impossible
  - we determine probability for each possible category
- Variability in feature values for objects in the same class versus those in different classes causes the difficulty of the classification problem
  - Variability in feature values may arise due to complexity, but also due to ***noise***
  - Noisy features and missing features are major issues

Bayesian Decision Theory

- A classifier's decision may or may not be correct, so setting should be probabilistic
- Probability distributions may be used to make classification decisions with least expected error rate



Bayesian Decision Theory

- ***Bayesian classifier*** classifies an object into the class to which it is most likely to belong, based on observed features
- Assume:
  - *a priori* probability $P(\omega_i)$ for each class ω_i
  - unconditional distribution $P(\mathbf{x})$
  - class conditional distribution $P(\mathbf{x}|\omega_i)$
- If all the classes are disjoint, by Bayes Rule, the *a posteriori* probabilities are given by:

$$P(\omega_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \omega_i) P(\omega_i)}{\sum_j P(\mathbf{x} \mid \omega_j) P(\omega_j)}$$

Bayesian Decision Theory

- If we have an observation x for which $P(\omega_1|x)$ is greater than $P(\omega_2|x)$, we would naturally prefer to decide that the true state of nature is ω_1
- Whenever we observe a particular x, the probability of error is

$$P(error \mid x) = \begin{cases} P(\omega_1 \mid x), \text{if we decide } \omega_2 \\ P(\omega_2 \mid x), \text{if we decide } \omega_1 \end{cases}$$

- Clearly, for a given x we can minimise the probability of error by deciding ω_1 if $P(\omega_1|x) > P(\omega_2|x)$
- The ***Bayes decision rule***

$$\text{Decide } \omega_1 \text{ if } P(\omega_1 \mid x) > P(\omega_2 \mid x); \text{otherwise decide } \omega_2.$$

Parametric Models for Distributions

- To compute $P(x|\omega_i)$ and $P(\omega_i)$, we can use an empirical method based on given samples
- Or if we know that the distribution of x follows a parametric model, then we may estimate the parameters using the samples
- *An Example*
  - Assume that the patterns in the r_{th} class can be described by a normal distribution, whose dispersion matrix Σ_r is known but the mean μ_r is unknown
  - Then, an estimate of the mean may be the average of the labelled samples available in the training set:

$$\overline{\mu} = \overline{x}$$

References and Acknowledgements

- Shapiro and Stockman, Chapter 4
- Duda, Hart and Stork, Chapter 1
- More references
  - Sergios Theodoridis, Konstantinos Koutroumbas, *Pattern Recognition*, 2009
  - Ian H. Witten, Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2005
- Some content are extracted from the above resources