The ultimate objective of many image analysis tasks is to discover meaning of the analysed image, e.g. categorise the objects, provide symbolic/semantic interpretation of the image or global understanding of the image scene.

Because these tasks are application specific no ready solutions are available. Hence, image analysis problems have strong resemblance to intelligent cognition rather than rely on application of “ready to use” set of standard processing techniques.
Intelligence:

⇒ ability to extract meaningful information from a set of irrelevant details,

⇒ capability to learn from examples and to generalise the acquired knowledge so that it can be useful in different processing contexts,

⇒ ability to comprehend data from incomplete information.
The concept of learning - example

Emergence
An example of a difficult pattern recognition problem.
Gestalt – Theory of mind and brain

Multistability

Reification (illusory objects)
Main parts of an image analysis system

- Preprocessing
- Image acquisition
- Segmentation
- Representation and description
- Knowledge processing
- Recognition and interpretation
- Analysed scene
- Result
Main parts of an image analysis system

- **Preprocessing**
  - **Low-level processing**
  - **Intermediate-level processing**
  - **Knowledge processing**
  - **High-level processing**

**Analysed scene**

Processing levels:
- Low-level processing
- Intermediate-level processing
- High-level processing

Result
A complete image analysis system comprises three major data processing modules:

**Low-level processing** - comprises image acquisition, image preprocessing and enhancement, e.g. noise reduction, contrast improvement, filtering, zooming, etc.

**Intermediate level processing** - deals with extraction and description of image components identified from a knowledge base, e.g. image segmentation, boundary and edge description, morphological processing.

**High-level processing** - involves classification, recognition and interpretation of the image scene.
Patterns and pattern classes

A **pattern** is a set of features that form a qualitative or structural description of an analysed object; in a mathematical sense a pattern is defined as a vector of features $\mathbf{x}=[x_1, x_2, \ldots, x_N]$.

A **pattern class** is a group of patterns that have similar feature vectors (according to some similarity measure). Pattern classes are denoted $\omega_1, \omega_1, \ldots, \omega_M$, where $M$ is the number of classes.

**Pattern recognition** (alternatively termed pattern classification) is the task of assigning patterns to their respective classes. It is equivalent to establishing a mapping:

\[ \mathbf{x} \mapsto \omega \]

from the **feature space** $X$ to the **pattern class space** $\Omega$. 
Pattern classification definition

feature space  pattern class space

many to one mapping

ω₁

ω₂

ω₃

ω₄

ω₅

ω₆
The mapping \( x \rightarrow \omega \) should point to the true class label \( \omega \).

The optimum among all the possible mappings \( x \rightarrow \omega \) is the one that generates minimum error rate.

This rate depends on the distribution characteristics of feature vectors in the pattern space \( X \), i.e., their distributions, overlap between distributions, etc.

If the knowledge about distributions of the patterns is unavailable the classification system must rely on the learning paradigms, e.g. \textit{artificial neural networks}. In such approaches the knowledge is acquired from a sufficiently large number set of training samples drawn from the pattern examples.
Selection of feature properties

- **discrimination** – features should assume considerably different values for patterns from different classes, e.g., diameter of fruits is a good feature for classification between grapefruits and cherries,

- **robustness** – features should assume similar values for all patterns belonging to the same class, e.g., colour is a poor feature for apples,

- **independence** – features used in a classification system should be uncorrelated, e.g., weight and size of a fruit are strongly correlated features,

- **small number of features** – complexity of data classification system grows substantially with the number of classified features, e.g., features that are strongly correlated should be eliminated.
Correlation coefficient between features $X$ and $Y$:

$$
\gamma_{xy} = \frac{1}{P} \sum_{i=1}^{P} \left( x_i - \mu_x \right) \left( y_i - \mu_y \right)
$$

where: $P$ – is the number of patterns, $\mu_x$, $\mu_y$ are the average values for features $X$ and $Y$ correspondingly, and $\sigma_x, \sigma_y$ are their standard deviations. The correlation coefficient assumes values in the range $[-1, 1]$. 

Selection of feature properties
Selection of feature properties

A measure of **separation** for feature $X$ between classes $j$ and $k$:

$$
\hat{D}_{xjk} = \frac{|\mu_{xj} - \mu_{xk}|}{\sqrt{\sigma^2_{xj} + \sigma^2_{xk}}}
$$

Large value for this measure means feature $X$ yields good separation between classes.
Reduction of the number of features

Direction of the first principal component

Angle $\theta$ chosen for best separation of features
Pattern classification problem can be approached by means of the so called decision functions. The number of these functions is equal to the number of classes. Let \( \mathbf{x} = [x_1, x_2, \ldots, x_N]^T \) represent an \( N \) dimensional pattern vector. For \( M \) pattern classes \( \omega_1, \omega_2, \ldots, \omega_M \), we are searching for \( M \) decision functions \( d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_M(\mathbf{x}) \) with a property that, if a pattern \( \mathbf{x} \) belongs to class \( \omega_i \) then

\[
d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \ldots, M; i \neq j
\]

In other words, decision function \( d_i(\mathbf{x}) \), “wins the competition” for assigning the input feature vector \( \mathbf{x} \) to the pattern class \( \omega_i \).
Decision functions

\[ d_2(\chi) \quad d_1(\chi) \quad d_3(\chi) \]

class region 2  class region 1  class region 3  class region 4
Decision functions

The decision boundary between two arbitrary classes \( i \) and \( j \) \((i \neq j)\) is defined by the function:

\[
d_{ij}(x) = d_i(x) - d_j(x) = 0
\]

Then, for patterns of class \( \omega_i \):

\[d_{ij}(x) > 0\]

and for patterns of class \( \omega_j \):

\[d_{ij}(x) < 0.\]
Minimum distance classifier

Assume that each pattern class is represented by a mean vector (also called a class prototype):

\[ m_i = \frac{1}{P_j} \sum_{x \in \omega_j} x, \quad j = 1, 2, \ldots, M \]

where \( N_j \) is the number of pattern vectors from class \( \omega_j \).

Possible way to determine the class membership of an unknown pattern vector \( x \) is to assign it to the class of its closest prototype vector.
Minimum distance classifier

If the Euclidean distance is used the distance measure is of the form:

\[ D_j(x) = \| x - m_j \| \quad j = 1,2,\ldots,M \]

and \[ \| x \| = (x^T x)^{1/2}. \]

Feature vector \( x \) is assigned to class \( \omega_j \) if \( D_j(x) \) is the smallest distance.
Minimum distance classifier

The following distance function can be constructed

\[ d_j(x) = x^T m_j - \frac{1}{2} m_j^T m_j \quad j = 1, 2, \ldots, M \]

and assigning \( x \) to class \( \omega_j \) if \( d_j(x) \) gives the largest value.
The decision boundary between classes $\omega_i$ and $\omega_j$ for a minimum distance classifier is:

$$d_{ij}(x) = d_i(x) - d_j(x) =$$

$$= x^T (m_i - m_j) - \frac{1}{2} (m_i - m_j)^T (m_i - m_j) = 0$$

The surface defined by this equation is the perpendicular bisector to the line joining $m_i$ and $m_j$. For $N=2$ the bisector is a line, for $N=3$ it is a plane, and for $N>3$ it is called a hyperplane.
Minimum distance classifier

\[ d_{12}(x) = 2.8x_1 + 1.0x_2 - 8.9 = 0 \]

\( (1.5 \ 0.3) \)
Example 1: Consider two classes denoted by \( \omega_1 \) and \( \omega_2 \), that have mean vectors \( \mathbf{m}_1 = (4.3, 1.3)^T \) and \( \mathbf{m}_2 = (1.5, 0.3)^T \). The decision functions for each of the classes are:

\[
\begin{align*}
    d_1(\mathbf{x}) &= \mathbf{x}^T \mathbf{m}_1 - 0.5 \mathbf{m}_1^T \mathbf{m}_1 = 4.3x_1 + 1.3x_2 - 10.1 \\
    d_2(\mathbf{x}) &= \mathbf{x}^T \mathbf{m}_2 - 0.5 \mathbf{m}_2^T \mathbf{m}_2 = 1.5x_1 + 0.3x_2 - 1.17
\end{align*}
\]

Equation for the decision boundary:

\[
d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 2.8x_1 + 1.0x_2 - 8.9 = 0
\]

Class membership of a new feature vector is defined on the basis of the sign of \( d_{12}(\mathbf{x}) \).
Voronoi mosaic

%MATLAB
rand('state',0);
x = rand(1,10); y = rand(1,10);
[vx, vy] = voronoi(x,y);
plot(x,y,'r+','vx,vy','b-');
axis equal;

- class prototype
Example 2:

Classification of American Bankers Association font character set.

The design of the font ensures that the waveform corresponding to each character is distinct from that of all others.

There are $N=14$ prototype vectors in 10-dimensional feature space.
The minimum distance classifier works well on the following conditions:

- the distance between class means is large compared to the spread of each class.
- classes are linearly separable (class boundaries are: lines, planes or hyperplanes).
Evaluation of pattern classification quality

Statistical distribution of single feature

- TN
- FP
- TP
- FN

decision border
**Evaluation of pattern classification quality**

### Confusion matrix

<table>
<thead>
<tr>
<th>Classifier output</th>
<th>True classification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class B (wrong, ill, …)</td>
</tr>
<tr>
<td>Class B</td>
<td><strong>TP</strong> (ang. true-positive)</td>
</tr>
<tr>
<td>Class A</td>
<td><strong>FN</strong> (ang. false-negative)</td>
</tr>
</tbody>
</table>
Evaluation of pattern classification quality

Accuracy:

\[ ACC = \frac{TP + TN}{TP + TN + FP + FN} \]

Sensitivity:

\[ SE = \frac{TP}{TP + FN} \]

All class B examples
Evaluation of pattern classification quality

**Specifity:**

\[
SP = \frac{TN}{FP + TN}
\]

**False alarm rate:**

\[
FA = 1 - SP = \frac{FP}{FP + TN}
\]

All class A examples

A

B

TN

TP

FP

FN
Receiver Operator Characteristic (ROC curve)

- False alarm rate
- Sensitivity
- Diagnostic parameter
- Good test
- Poor test

See excellent demo explaining ROC:
http://www.anaesthetist.com/mnm/stats/roc/Findex.htm
Types of classifiers

**Parametric**: probability density functions are known and only parameters of these functions are unknown e.g., the Gaussian distribution is defined by its mean $\mu$ and standard deviation $\sigma$), The Bayes classifier is a parametric classifier, i.e., requires knowledge about density function distributions

**Nonparametric**: probability density functions are not known and must be estimated from a sufficiently large measurement data.
Image segmentation example
Colour based segmentation \{R,G,B\}

Statistical distribution of colour components for the indicated image regions
The Bayes classifier

$c_1$ - class 1
$c_2$ - class 2

Probability density

$p(x)$
$p(x/c_1)$
$p(x/c_2)$

$a$ priori probabilities

$m_2$
$x_0$
$m_1$
The Bayes classifier

From the probability theory the following holds:

\[ p(a / b) = \frac{p(a) p(b / a)}{p(b)} \]

hence:

\[ p(a / b) = \frac{p(a) p(b / a)}{p(b)} \]

\[ p(c_i / x) = \frac{P(c_i)p(x' / c_i)}{p(x)} \]

**Bayes rule**

*a priori probability*

*a posteriori probability*
An image pixel is assigned to the class for which:

$$L_i = \arg \max_i \left\{ p(c_j / x) = \frac{P(c_i)p(x / c_i)}{p(x)} \right\}, \quad i = 1,2,\ldots,N$$

i.e.

$$L_i > L_j, \quad j \neq i, \quad i = 1,2,\ldots,N$$
The Bayes classifier

$c_1$ - class 1
$c_2$ - class 2

$p(x)$

$p(x/c_2)$

$p(x/c_1)$

$p(c_2/x) > p(c_1/x)$

$p(c_1/x) > p(c_2/x)$

$a$ priori probabilities

$m_2$

$x_0$

$m_1$
The Bayes classifier

A priori probability for 1D Gaussian distribution:

\[ p(x / c_i) = \frac{1}{\sqrt{2\pi \sigma_i}} \exp \left[ - \frac{(x - m_i)^2}{2\sigma_i^2} \right] \quad i = 1, 2 \]

A posteriori probability:

\[ p(c_i / x) = p(x / c_i)P(c_i) = \frac{1}{\sqrt{2\pi \sigma_i}} \exp \left[ - \frac{(x - m_i)^2}{2\sigma_i^2} \right] P(c_i) \quad i = 1, 2 \]
The Bayes classifier - example

We pick a pot randomly and then the ball randomly. **Question:** From which pot the ball was picked?
Before we see the colour of the ball we picked (a priori knowledge) we assume the probability of choosing each of the pots is equal, i.e. 0.5.
The Bayes classifier - example

Suppose we picked a red ball. Can we verify our first hypothesis having this a posteriori knowledge? Yes, the answer comes from the Bayes theorem.
The Bayes classifier -example

Pot A

- 20 – red
- 20 – white

Pot B

- 30 – red
- 10 – white

That was the pot!
probably

$p(A/\text{red}) = \frac{P(A)p(\text{red}/A)}{P(A)p(\text{red}/A) + P(B)p(\text{red}/B)} = \frac{0.5 \times 0.5}{0.5 \times 0.5 + 0.5 \times 0.75} = 0.4$

$p(B/\text{red}) = \frac{P(B)p(\text{red}/B)}{P(A)p(\text{red}/A) + P(B)p(\text{red}/B)} = \frac{0.5 \times 0.75}{0.5 \times 0.5 + 0.5 \times 0.75} = 0.6$
Multivariate Gaussian distribution

A priori probability for 2D Gaussian distribution $x^T=[x_1 \ x_2]$: 

$$p(x / c_i) = \frac{1}{(2\pi)^{1/2}\left|\sum_i\right|^{1/2}} \exp \left[ -\frac{(x - m_i)^T \sum_i^{-1} (x - m_i)}{2} \right] \quad i = 1, 2, \ldots, N$$

Where $i$ is class index of $N$ given classes and the covariance matrix:

$$\Sigma_i = \begin{bmatrix} \sigma^2_{x1} & \sigma_{x1}\sigma_{x2} \\ \sigma_{x2}\sigma_{x1} & \sigma^2_{x2} \end{bmatrix}$$
2D Gaussian distribution - example

Let \( m_1=m_2=0 \):

\[
\Sigma = \begin{bmatrix} 3 & 2 \\ 2 & 4 \end{bmatrix} \quad \rightarrow \quad \Sigma^{-1} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{4} \\ \frac{1}{4} & \frac{3}{8} \end{bmatrix} \quad \rightarrow \quad |\Sigma| = 8
\]

\[
p(x) = \frac{1}{(2\pi)^{1/2}} \exp \left[ -\frac{1}{2} (x^T \Sigma^{-1} x) \right] = A \exp(B)
\]

\[
A = \frac{1}{(2\pi)^{1/2} 8^{1/2}} = \frac{1}{4\sqrt{\pi}}
\]

\[
B = -\frac{1}{2} \left( \frac{1}{2} x_1^2 - \frac{1}{2} x_1 x_2 + \frac{3}{8} x_2^2 \right)
\]
Colour based segmentation \{R,G,B\}

Multivariate Gaussian distribution
Selection of the feature vector

Feature vector:

$$\theta = \{x, y, R, G, B, u, v\}$$

- coordinates
- colour
- optical flow

$$u = \frac{dx}{dt}$$

$$v = \frac{dy}{dt}$$
Optical flow features

Optical flow based segmentation \( \{u, v\} \)
In each point of the feature space, a number of $k$ neighbours is counted. The point is assigned to the class from which there is the largest number of points in the neighbourhood.
Image segmentation examples
Image segmentation examples
Fingerprint recognition

FBI image database of fingerprints (1992)
Iris recognition (biometry)

The patented Daugman’s algorithm
Analysis of biomedical images
Image database querying

„Idea“ of the search image or a copy of the image database hit

DWT

Analysis of X-ray images of grains

Automatic text detection