How to build
pattern recognition systems

Lesson 1
9 April 2007

Matteo Roffilli

roffilli@csr.unibo.it
http://www.cs.unibo.it/~roffilli
Prerequisites

1. Available time for 1 image/frame/etc
2. Available space for hardware: PC, cluster, mainframe, embedded, etc
3. Available budget for hardware: $$$
   a) Available memory: 128 MB, 1 GB, 1 TB
   b) Available processor
4. Available time for developing
5. Available know-how of the developers
Medical imaging PR system

1. 10 sec / image
2. PC
3. 1000 $
   a) 1 GB RAM
   b) 3 GHz single/dual processor
4. Medium (1-3 years)
5. High-level
WHAT to do
Computer Aided Detection

CAD GOAL: to aid the radiologist in detecting tumoral masses
1. to classify unknown images in 2 classes: diseased or healthy
2. to locate the lesion

REQUIREMENT: to find all lesions without prompting false signals
State of the art: Detection

1. Based on appearance models (external knowledge)
2. Segmenting borders is a difficult task and some difficult masses are lost!
3. About **10-50** Regions Of Interest (ROIs)
Convolution filter
State of the art: Classification

Data representation
Feature extraction

- Area
- Perimeter
- Size
- Intensity
- Shape
- …

About 10-15 features

Classification

- ANN
- RBF
- Bayesian Networks
- Decision Tree
- Hand-made classifiers
State of the art: Result

ROIs classified as positive are prompted on the original image
A novel brute force approach

The novel contributions of our work are mainly three:

1. the detection step is performed without the use of external knowledge

2. the feature extraction step is avoided

3. SVM and RVM are used as classifiers for the classification step
Detection

This detection produces about **100,000 ROIs**
HOW to do
Computational overview

• Red blocks produce about 80% of computational cost.
• How can we improve the performance?

Efficient exploitation of memory/CPU bus!!!
Data representation

- pixel
- wavelet
- ranklet

This vector identifies a point in a \textit{n-dimensional space} (n \sim 4000)
Each element of the vector is a \textit{feature}
Memory access

Not efficient

Efficient

Matrix float**

Array float*
Data representation storage

~ 100’000 * 4000 float values
~ 4*4*10^8 bytes
~ 1.50 Giga bytes
Data representation storage

Efficient

~ 100’000 * 4000 float values
~ 4*4*10^8 bytes
~ 1.50 Giga bytes

Only ASM/C/C++ can deal with efficient memory management !!!
The C supremacy

1. Efficient code
2. Efficient compiler
3. Efficient linker
4. ANSI C is portable: Linux, Windows, etc
5. ANSI C will be always available (e.g. VB6 is dead!)
6. *ALL* O.S. are written in C
7. C can produce .SO/.DLL to be called by others
8. C is simple for algorithm implementation
9. By using C you can access SIMD instructions
Next lesson…