Web tools to support Image Classification

F. Odone, A. Barla, E. Franceschi, A. Verri

National Institute of Solid State Physics, Genova, Italy
Università di Genova, Italy

Statistical Learning & Image Processing slipGURU Genova University Research Unit

http://slipguru.disi.unige.it

How we use the Web to collect data for our research on image classification

F. Odone, A. Barla, E. Franceschi, A. Verri

National Institute of Solid State Physics, Genova, Italy
Università di Genova, Italy

Statistical Learning & Image Processing slipGURU Genova University Research Unit

http://slipguru.disi.unige.it

Summary

- Motivations

- Automatic images download
- Automatic image filtering

- Run time experiments
- Conclusions

Motivations

- Image understanding (IU) is far from being solved
- Decades of computer vision methods do not seem to help much

Let's face it: image understanding is *AI-complete*

Motivations

- Image classification as way to annotate images automatically
- We address the problem in the old fashioned way: build a dictionary of classifiers
- We are aiming at a high number of classifiers (hundreds?) describing general concepts (indoor-outdoor, day-night, sunsets, landscapes, cities..) or finding dominant things (sea, sky, grass..)

Motivations

Why do we think this time it might be working?

- Statistical learning advances
- The available computing power makes it possible to try many things

Our ingredients:

- Statistical learning tools
- (mostly) classic computer vision techniques to express image similarity

What do we need

- A well known practical problem with example based methods is that for each classification problem a suitable set of training data needs to be provided.

- The growth of the Web provides us with a potentially huge source of data --- images in our case.

- Images on the Web are of various sorts, not always useful or interesting. They should be classified...chicken and egg problem!

List of wishes

- Automatic download of many images
- Random selection of the origin
- Low level filters: ability to control size, format, degree of compression
- High level filters: some capabilities of grouping homogeneous data, e.g. photos, monochrome images, computer generated graphics..

Image download

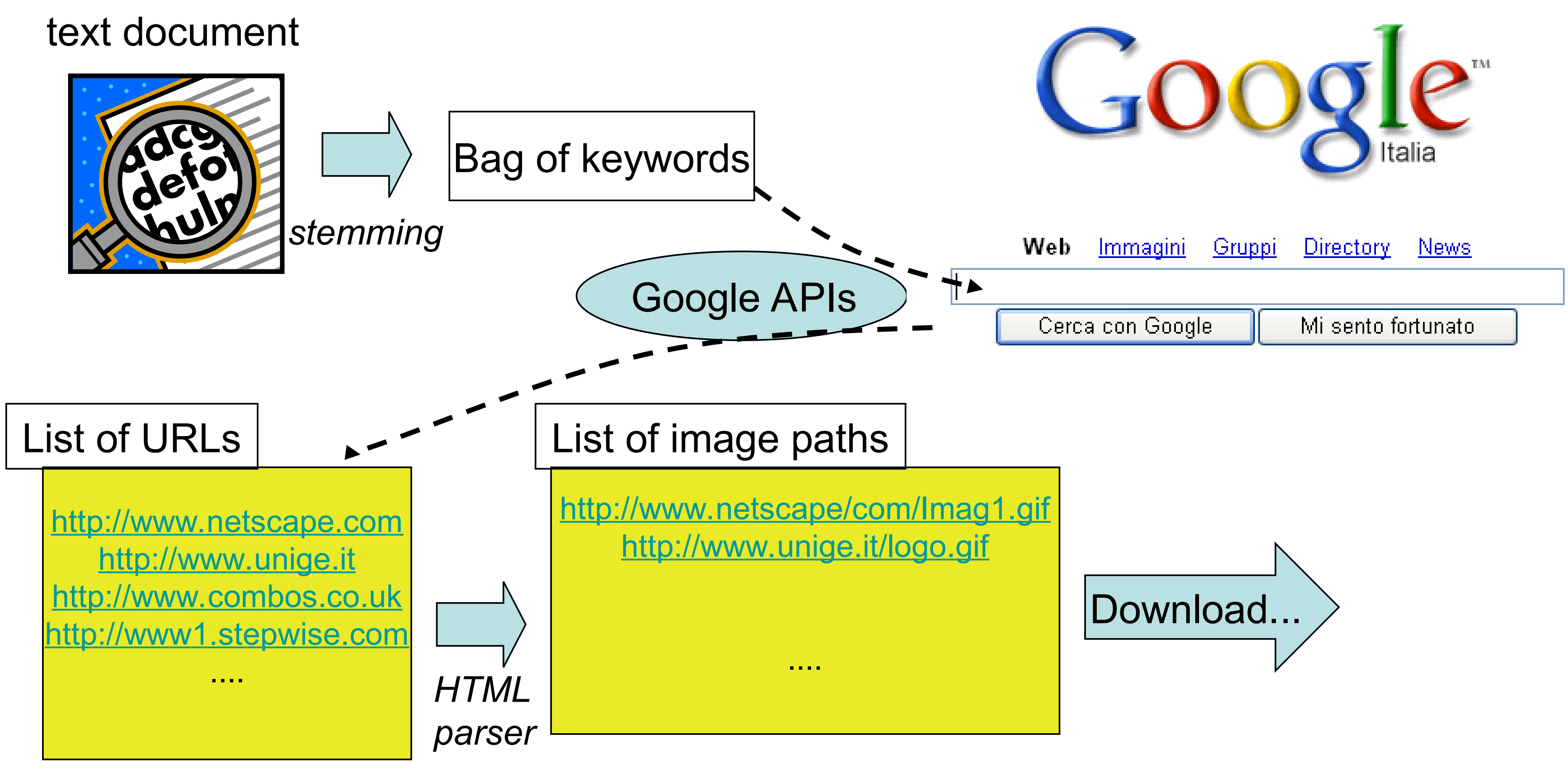
- *C.Ra.W.L.er* downloads images from the Web automatically and performs (offline) a first stage of filtering by content

- It is based on the idea that, thanks to the size of the Web, if one chooses *a high number of random nodes and jumps among them* it is virtually impossible to go back on the same path.

Images download

- In order to get the nodes we feed the system with a random text document of reasonable length
- The system extracts all words and uses them as keywords for a text search engine (Google)
- The *Google Web APIs service* offers a maximum of 1000 daily searches giving 10 links back.

Images download



Images download

- The main feature of this algorithm is that it downloads a small percentage of duplicate images and doesn't fall into loops

- In our experiments we performed a simple operation to check for duplicates, based on comparing mean color values, and found that only the 6% of images are potential duplicates
- In many cases they are different sized images.

Filters based on measurements

- Aspect ratio: checking the aspect ratio is a simple way to control the number of banners

- Color histogram analysis in the RGB space:
  - BW: most pixels fall in the black and white bins
  - Gray-level images: most pixels RGB values are (x,x,x), x in (0,1)

Filters based on classifiers

- Color against monochrome:
  - It is a generalization of the previous filter that considers a more general family of monochrome images
  - We use a SVM classifier trained on a small but carefully selected training set of data (100 images)
  - The image description is a simple color histogram


# of bins	RGB	HSV
4x4x4	78.3 ± 0.6	78.6 ± 0.8
8x8x8	81.6 ± 0.8	78.6 ± 0.9
16x16x16	98.3 ± 0.5	98.5 ± 0.5
20x20x20	98.2 ± 0.4	98.9 ± 0.2

Validation sets: 400 color images (both photos and graphics) 200 monochrome images

Filters based on classifiers

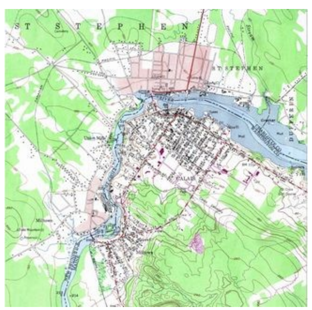
- Natural photos against other images:
  - the Web is populated with various sorts of synthetical images
  - Most computer vision and image classification assume that images are natural photos; such images are a minor part of the population of pictures of the Web
  - A human observer classifies easily artificial photos

- The problem of classifying photos from other kind of pictures automatically is not trivial and it has been often addressed (Athitsos *et al* 1997, Smith *et al* 1997, Prabhakar 2002)

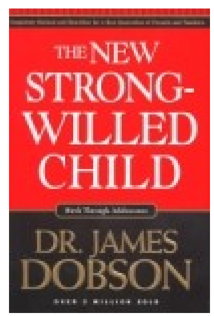
Photos and synthetic images

- Part of the problem is related to the definition of negative examples:
  - What are the pictures that we call synthetic images?

Scanned hand drawing



Digital maps



Scanned graphics

Digital cartoons

Digital synthetic photos

Photos and synthetic images

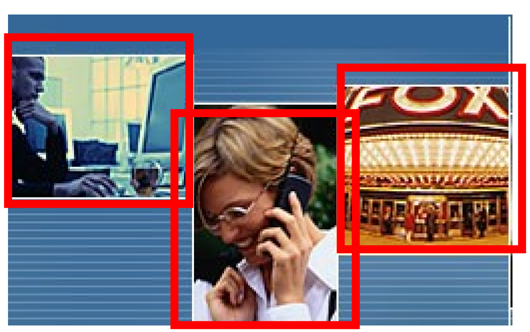
- Data sets of 420 photos and 420 non photos:
  - Split in various training+validation of 120+300 images each
  - We compare a number of fast to compute image representations mostly based on color (color histograms on different color spaces) and texture (gradient histograms, edge lengths…)
  - We train a binary SVM equipped with the histogram intersection kernel (Barla *et al*, 2003)

Photos and graphics

Image representation	Recognition rate
HSV color histogram (4x4x4 bin)	66.1 ± 0.4
RGB color histogram (4x4x4 bin)	67.3 ± 0.5
HSV color histogram (16x16x16 bin)	86.1 ± 0.2
RGB color histogram (16x16x16 bin)	88.4 ± 0.3
Histogram of edge points strength	89.4 ± 0.4
Histogram of edge directions	49.4 ± 0.4
Histogram of edge chains lengths	84.1 ± 0.5
HSV histogram + edge strength hist.	93.2 ± 0.3
RGB histogram + edge strength hist.	93.8 ± 0.2
HSV histogram + edge length hist.	93.2 ± 0.9
RGB histogram + edge length hist.	93.3 ± 0.8

Open issues (current work)

- Digitally retouched photos (Photoshop…)
  - Results are unpredictable, depending on the quantity and type of intervention
- Mixed images:
  - local analysis with multiscale local features



?

Run time experiments

- We have run the system on 10 times on different texts as seeds.
- The maximum time required was 8 hours on with a busy network. The median time is 3 hours

- We discarded small images (smaller than 100x100 pixels), kept all formats, did not check aspect ratio

Run time experiments

Image classes	# detections (of which mixed images)	#false positives (of which mixed images)
B/W	17 (0)	1 (0)
Monochrome	16 (0)	2 (0)
Graphics	2454 (289)	271 (108)
Natural photos	1304 (186)	193 (76)

After having discarded manually the mixed images we obtain a recognition rate of 93,6 % on natural photos and 89.4% on graphics

If we are interested to one specific class we may control the number of false positives

Conclusions

- *C.Ra.W.L.er* allows us to download images from the Web automatically and perform a first stage of filtering by content
- The search starts from a text the semantic content of which may be related to a subject or not.
- The download phase is still quite slow compared to the number of images downloaded but the number of duplicate images is small
- The classifiers implemented so far achieve satisfactory results
- An interface for helping final manual labeling is under development

The end!

Learning from examples

- We say that a program for performing a task has been acquired by learning if it have been acquired by means other than explicit programming (Valiant 1984)

- We assume that all we have is a set of examples and look for inference schemes that allow us to model the data in a predictive (generalizing) way.
- We say that our model has generalization properties if it describes new examples not belonging to the initial set of data.

Statistical Learning
(Vapnik 1995, ...)

- Given $\{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$, drawn i.i.d. from a fixed but unknown probability distribution, the problem is to find a function $\hat{f}$ minimizing the functional

$$\min_f \frac{1}{\ell} \sum_{i=1}^{\ell} V\big(f(x_i), y_i\big) + \lambda \|f\|_K^2$$

Statistical Learning

- The solution (optimal trade-off between the data term – or empirical error, and the prior – or regularization term) takes the form

$$\hat{f}(x) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x)$$

where the coefficients α_i are determined by the examples and are bounded by the inverse of the regularization parameter λ.

What do we see?

stuff

stuff

stuff

object

Outdoor photo

Don’t forget that…

- A system learns what we teach it
- Once we choose a data representation the damage is done

Image classification in Genova

A. Classification of global properties based on low level representations of images and statistical learning;
   e.g. Is it an indoor image? Is it a sunset?

B. Example-based object detection with hypothesis tests;
   e.g Where are the faces in this image?

C. Stuff detection as a combination of low-level information and feature selection;
   e.g. What sort of “stuff” do I see in this image?

D. Location and classification of dominant objects via local descriptors.
   e.g. How do I describe an object using local keypoints?

Image classification from low-level global description

(Barla *et al* 2003, Odone *et al* 2005)

- The problem:
  - Capture a global description of the image using simple features
- The procedure:
  - Build a suitable training set of data
  - Find an appropriate representation
  - Choose a classification algorithm
  - Tune the parameters and select a similarity function

Computer vision ingredients

We represent whole images with low level descriptions of color, shape or texture

- Color:
  - Color histograms
- Shape:
  - Orientation and strength edge histograms
  - Histograms of the lengths of edge chains
- Texture:
  - Wavelets, Co-occurrence matrices

The choice of the kernel

- The prior knowledge on the problem domain can (and should) be used for constructing appropriate kernels because
  - it allows for learning from fewer examples
  - it minimizes the preprocessing requirements
  - in some cases it is simpler than using off-the-shelf kernels (e.g., in terms of parameter choice)

Example: histogram intersection kernel

- Histogram Intersection is a powerful similarity measure for color indexing
- Given two images, *A* and *B*, of *N* pixels, if *M* is the number of bins, histogram intersection is defined as

$$K(A,B)=\sum_{i=1}^{M}\min\{A_i,B_i\}$$

- It can be shown (Barla *et al.*, 2003) that histogram intesection is a Mercer's kernel

A few examples and results

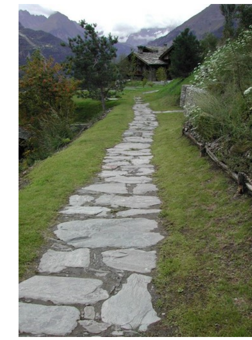
Indoor-outdoor classification

- HSV color histograms with 20x20x20 bins
  (best representations among the ones tried…)
- Binary SVM with histogram int. kernel
  (that outperformed standard kernels)

- Training set of about 800 images
- Validation set of about 1500 images
- Recognition rate: 89.7 ± 0.3
- online version trained on about 5000 images: http://slipguru.disi.unige.it

A few examples and results

Cityscape retrieval

- HSV color + edge directions histograms
  (best representations among the ones tried…)
- Binary SVM with histogram int. kernel
  (that outperformed standard kernels)

- Training set of about 800 positive and negative images
- Validation set of about 900 positive and negative images
- Precision: 85.1 ± 1.7 Recall: 90.1 ± 1.7

A few examples and results

Graphics - Photos

- RGB color + edge strength histogram
  (best representations among the ones tried...)
- Binary SVM with histogram int. kernel

- Training set of about 300 images
- Validation set of about 500 images
- Recognition rates: 93.2 ±0.3
- At run time, over 1000 images automatically downloaded from the Web 86.9% r.r. on graphics and 94.6% on photos

Image classification from low-level global description