UNIVERSITÀ
DEGLI STUDI
FIRENZE

Scuola di Ingegneria
Corso di Laurea Magistrale in Ingegneria
Informatica

CNN for Automatic Data Collection and Filtering

Basi di Dati Multimediali

Saverio Meucci

Anno Accademico 2015/2016

Introduction

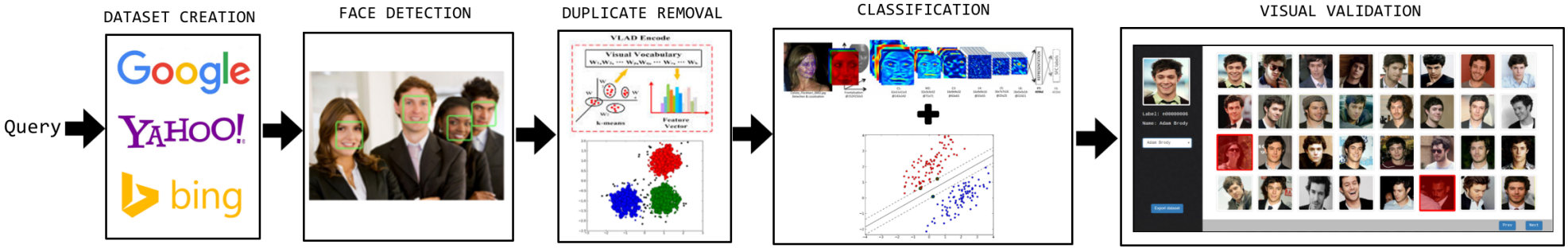
Objective: the creation of a procedure, based on the work of [1], thanks to which build a large dataset of images of faces that can be used in the training process of a **CNN**.

This procedure takes advantage of the modern **search engines**, and, at the same time, has the objective of decreasing the cost in terms of human time necessary to the acquisition and annotation of the dataset.

Introduction

The proposed method is divided in phases, as follows:

- A **dataset creation** phase, that exploits modern search engines.
- Detect the faces inside the downloaded images thanks to a **face detector**.
- A **duplicate removal** phase, thanks to which duplicate images are removed.
- A **classification** phase, to determine if the faces are correctly associated to an identity.
- A **visual validation** phase, done by a user with the help of a web application.



Dataset Creation

The phase for the **dataset creation** consists in the acquisition of images exploiting the modern *image search engines*.

It is necessary to decide a list of **identities** that are sufficiently popular, so that for each identity there will be a high number of images (between 500 and 1000), as results of the search engines.

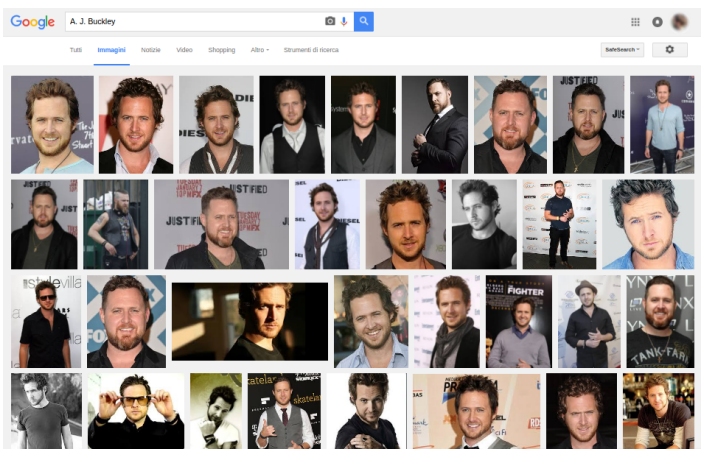
The acquisition procedure is divided in two parts:

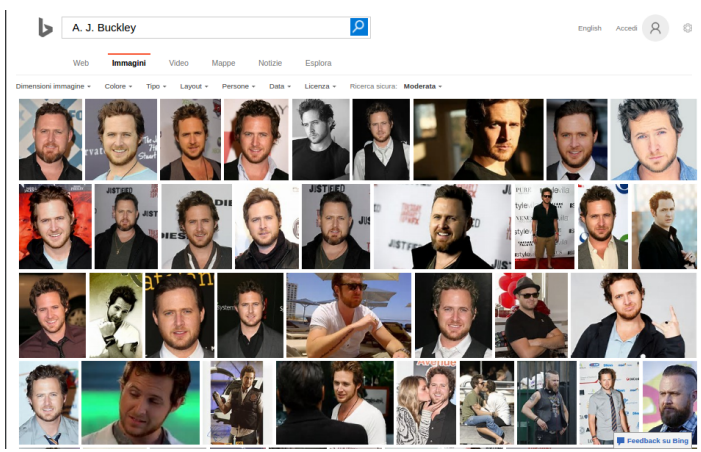
- A collection phase.
- A download phase.

Dataset Creation

The **collection** phase takes advantage of three search engines, such as Bing, Yahoo and AOL, to query and obtain images for each identity.

A query to an image search engine returns a HTML page; thanks to a parser, implemented for each search engine, we can obtain the links to the images contained in each HTML page.





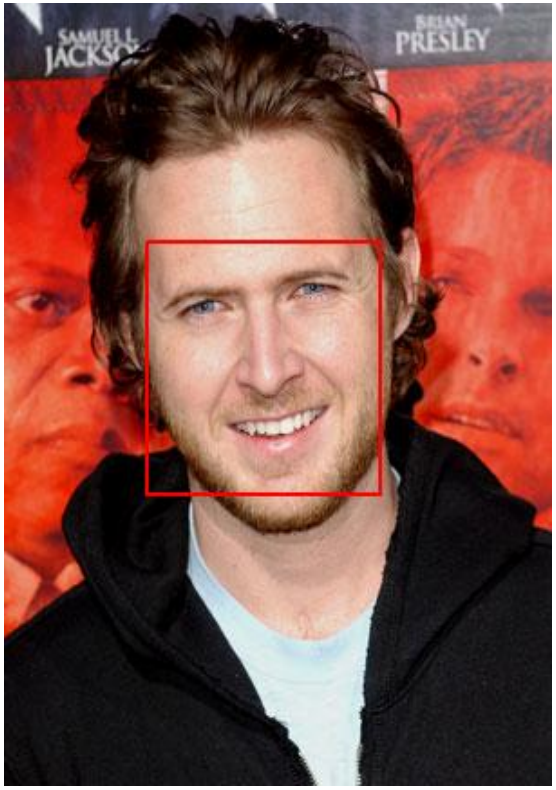
The **download** phase is straight forward: the images are downloaded and saved in a folder related to the identity, using the links obtained in the previous step.

Face Detector

The next phases have the objective of **filtering** the obtained dataset, in order to remove, for each identity, the images that are not relevant.

The results of the queries to the image search engines for a specific identity, will also contain images that are not related to that identity, thus the need of a filtering phase.

Since the objective of this paper is the creation of a dataset of images of **faces**, we need to detect and select the faces in the retrieved images.

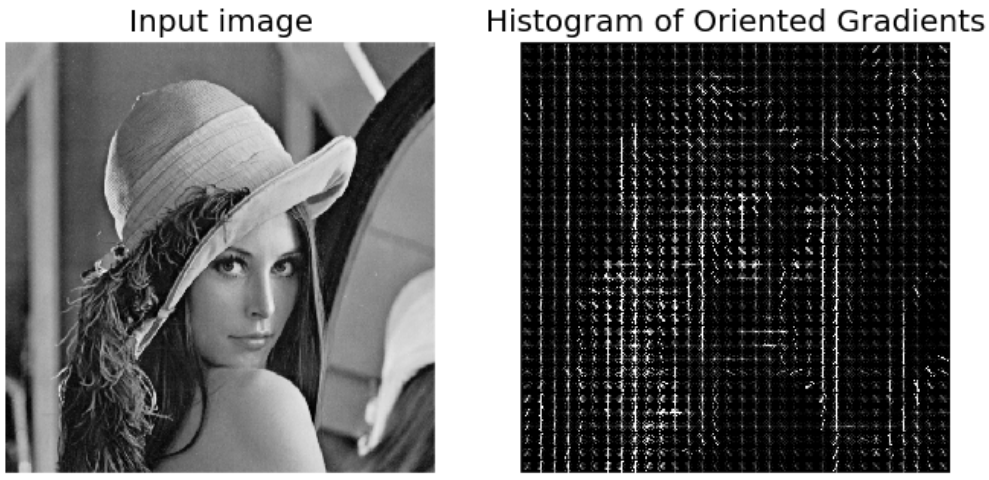


Face Detector

This task is accomplished by implementing a **face detector**, written in C++ using the **Dlib** library [2].

The **Dlib** C++ library implements a face detector by using the classic **histogram of oriented gradients** (**HOG**) [6] feature combined with a linear classifier, an image pyramid and sliding window detection scheme.

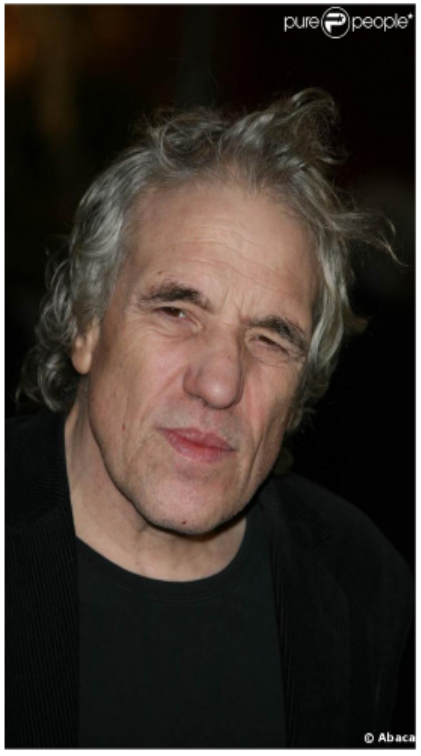
The face detector takes an image, or list of images, and returns the coordinates of the **bounding boxes** surrounding the detected faces.

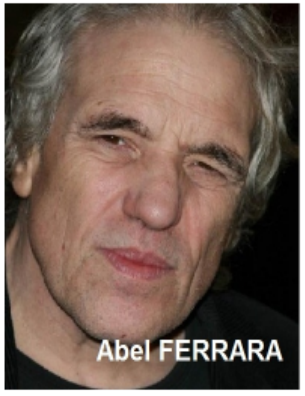


Duplicate Removal

In the results of an image search engine there will be images that are **duplicates**, both because the returned links refer to the same location and because the same image is stored in two different locations.

That is not desirable, since the dataset will be used to train a **CNN**. A training phase need to generalized as much as possible the objects that is learning; a duplicated image does not add any useful information to this process.



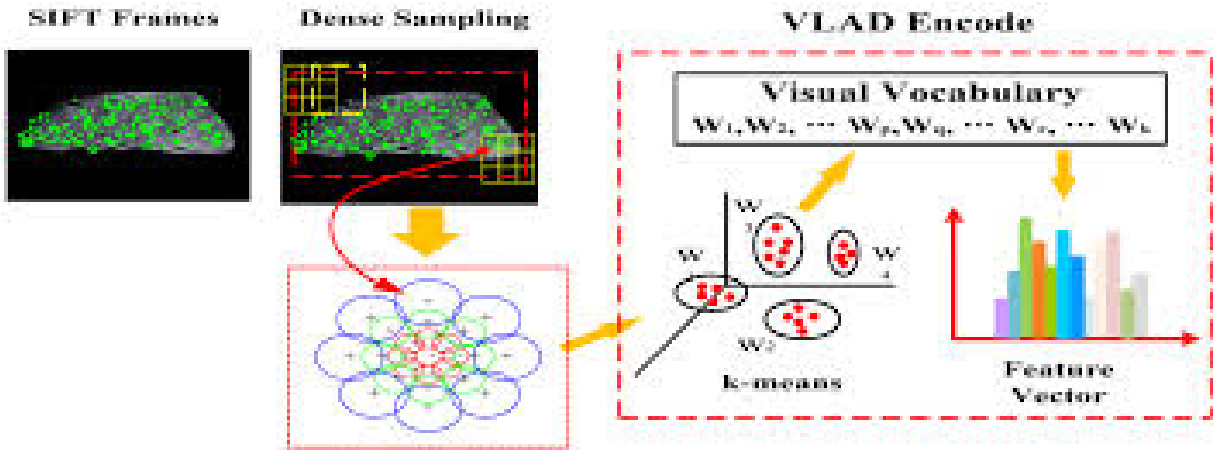


Duplicate Removal

For each image, a **feature vector** is computed using the **Vector of Locally Aggregated Descriptor** (**VLAD**) encoding, using the implementation from the **VLFeat** library [5].

The **VLAD** is a feature encoding and pooling method that encodes a set of local feature descriptors extracted from an image using a feature dictionary built using a clustering method.

These feature vectors are then **clustered** within the images for each identity; only a single element per cluster will be retained in order to removed the duplicated images.



Classification

It is necessary a **classification** phase in order to clean the dataset from the images that are not relevant to the associated identity.

We consider as images the bounding boxes extracted by the face detector, in order to focus the classification to the detected faces.

Taking advantage of a **pre-trained CNN** of faces from 2622 identities, we extract from the net the last layer of convolution as feature vector for each images in the dataset. For each identity, a **linear SVM** will be trained.

The feature vectors were extracted by the net using the **MatConvNet** toolkit [4] and the model was trained using the **LIBSVM** library [3].

Classification

In order to reduce the computational complexity, each identity is trained against other K (e.g. $K = 5$) identities, taking 50 images from each one.

The images that are used in the training phase were chosen by taking into account the **ranking** of the search engines, assuming that the higher the ranking the more relevant the images.

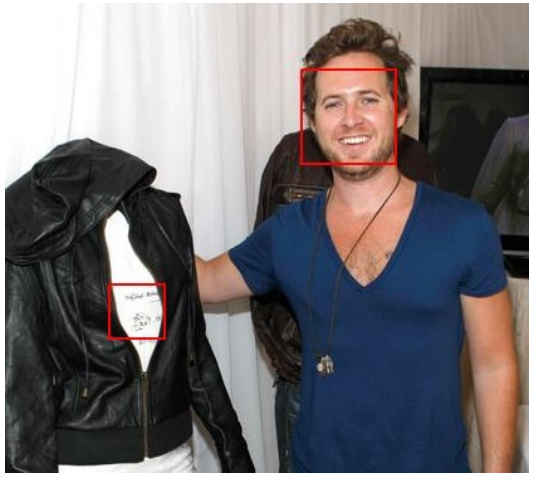
It is a **binary** classification for each identity and so we need to train each identity separately.

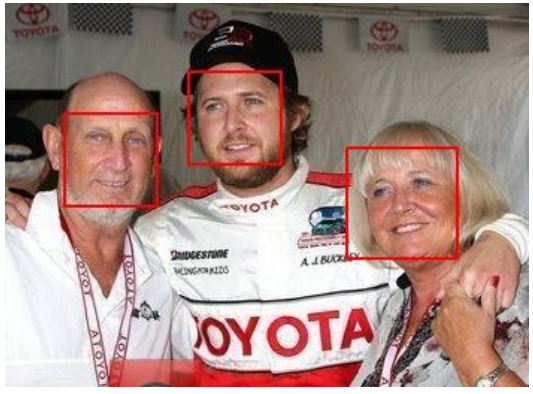
Validation

The set of faces used in the training of the models might not be completely clean:

- If an image contains two faces, each of them will have the same ranking; thus, if the ranking is high, they will both be used in the training phase but only one of them will correctly represent the identity.
- The same can be said about false positive detection.

Since it is not possible to resolve these problems in the previous steps, we have classification models that have been obtained with training sets containing errors.

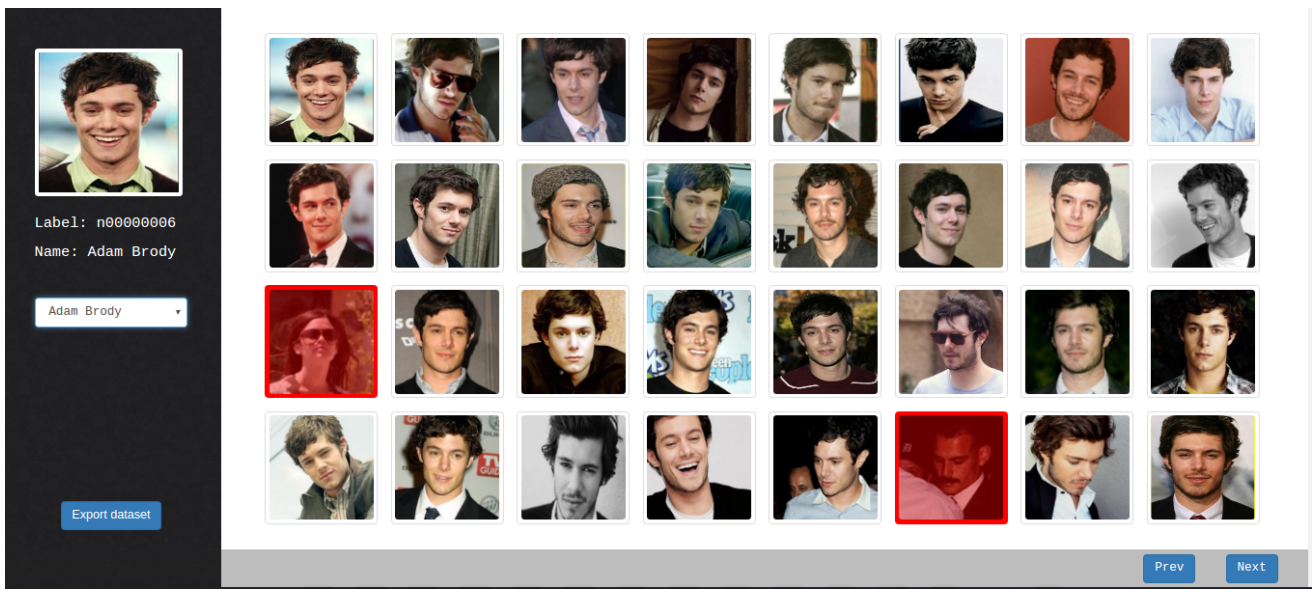




Validation

A **web application** for **visual validation** has been created for this purpose, thanks to which are shown to a **user** the images associated to each identity along with the results of the classification.

A **user** can interact with the web application, by browsing each identity and the images that are shown in a gallery; the user can also double click an image to **correct** the results of the classification. A user can also decide to completely remove an identity from the dataset.



Results

To evaluate the performance of this procedure, **experiments** have been performed with two different datasets.

One dataset (test A) contains 50 identities, such as **celebrities** and **actors**, the other (test B) contains 50 identities, such as **football players**.

The identities for the first dataset are a selection of the ones used by the pre-trained CNN that was used to extract the feature vectors for the images.

Results

Using the **ground-truth** created by the validation phase, it was possible to compare the results of the automatic classification.

Test	TPR	TNR	FPR	FNR	Accuracy
A	0,993	0,874	0,126	0,007	0,973
B	0,978	0,839	0,161	0,022	0,952

The average number of images per identity for the **prediction** is 420 for test A and 456 for test B; the average number of images per identity after the **validation** is 412 for test A and 448 for test B.

Conclusions

The implemented procedure presents some limitations:

- The collection phase is not reliable.
- The duplicate removal phase is only able to remove perfect duplicate.
- Having more than one detected face in an image can lead to a training set for the classification that contains errors, negatively affecting the computed model.
- Some identities can lead to ambiguous results in the image search engine.

References

[1] O. M. Parkhi, A. Vevaldi, A. Zisserman, Deep Face Recognition, British Machine Vision Conference, 2015.

[2] Davis E. King, Dlib-ml: A Machine Learning Toolkit, Journal of Machine Learning Research, 2009, 10, 1755-1758.

[3] Chang, Chih-Chung and Lin, Chih-Jen, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent System and Technology, 2, 3, 2011, 27:1—27:27.

[4] A. Vevaldi and K. Lenc, MatConvNet: Convolutional Neural Networks for MATLAB, Proceedings of the ACM Int. Conf. On Multimedia, 2015.

[5] A. Vevaldi and B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008.

[6] N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005.