Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Eye detection and face recognition in watch-list applications

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Alejandro Ariza Bueno

DIRECTORS: Francesc Tarrés / Raúl Quijada

DATE: February 2014

Títol: Eye detection and face recognition in watch-list applications

Autor: Alejandro Ariza Bueno

Directors: Francesc Tarrés / Raúl Quijada

Data: Febrer 2014

Resum

Els humans tenim una capacitat innata pel reconeixement d'altres persones a través de la cara independentment de les modificacions que aquesta pugui sofrir ja sigui a través de l'edat o essent camuflada per unes ulleres o una barba. En aquesta tesi es vol intentar desenvolupar un sistema complert de reconeixement facial a través d'un ordinador, es a dir, implementar un sistema autònom capaç de reconèixer persones d'una base de dades a partir d'una imatge desconeguda pel sistema i obtinguda de qualsevol font dins d'Internet (Twitter, Facebook, Google, etc.).

La idea doncs és per una banda crear una base de dades (que anomenarem watchlist) a partir d'imatges d'Internet suficientment potent i poder així entrenar a un reconeixedor amb la suficient fiabilitat que ens permeti obtenir un grau alt d'èxit de reconeixement. I d'altra banda, poder ser capaços de reconèixer el major numero d'imatges possible que trobem a Internet.

Per assolir el nostre objectiu s'ha implementat un sistema d'adquisició i processat d'imatges que adaptarà les fotografies extretes d'Internet perquè el sistema de reconeixement funcioni correctament. D'aquesta manera no és necessari generar una nova base de dades mitjançant la realització de fotografies específiques a la persona que volem reconèixer, ja que en la majoria dels casos això no ens serà permès.

Així doncs, s'ha utilitzat un codi ja implementat pel reconeixement facial aprofitant els recursos que ofereix OpenCV [1] i que ens ajudarà a testejar l'eficiència de la watchlist a la vegada que s'ha implementat un detector de cares i ulls juntament amb una sèrie d'algoritmes que milloren l'eficiència de la detecció capaç de normalitzar una imatge de manera automàtica. El resultat, aplicat a diferents imatges de diferents persones, serà la nostra watchlist i servirà per entrenar el reconeixedor i reconèixer noves cares. L'eficiència d'aquesta watchlist automàtica serà contrastada amb una altra watchlist creada manualment on l'usuari marcarà la posició dels ulls.

El resultat és un sistema complert de reconeixement facial totalment automàtic amb un determinat percentatge d'èxit de reconeixement. En treballs futurs es pot preveure la seva optimització i la implementació de casos pràctics.

Títol: Eye detection and face recognition in watch-list applications

Author: Alejandro Ariza Bueno

Directors: Francesc Tarrés / Raúl Quijada

Date: February 2014

Overview

Humans have a strong capacity for the recognition of other persons through their faces despite of modifications like aging or being covered by some glasses or beard.

This thesis develops a complete face recognition system implemented by a computer, what it means to build an autonomous system able de recognize persons from a database taking into account an unknown input image. Those images are taken from any internet source (Facebook, Twitter, Google...).

The aim of this thesis is to build a powerful database (watchlist named) from internet pictures used to train a recognizer in order to achieve a high recognition successful rate, and to define a system able to recognize as many pictures as possible obtained from internet.

To reach our goal it has been implemented an acquisition and processing image system to adapt those internet pictures due to make the system able to work properly. In this way, it was not necessary to create a new and specific database based on pictures taken directly from the person we want to recognize as in most cases this procedure cannot be followed.

Therefore, the system proposed is based on programmed sources provided by OpenCV [1] in combination with algorithms implemented in order to detect faces and eyes due to normalize the images taken from internet. The result, applied to several images from different characters will be the watchlist for training the recognizer. Finally, the proposed system is able to create a watchlist automatically. However, in order to validate the suitability of the method proposed, it is compared with another watchlist manually designed, where the user marks the position of the eyes.

The conclusion obtained is the suitability of the system proposed in this thesis as an automatic tool for recognizing faces and eyes with a determinate success rate in recognition. In the future works it has been thought the optimization of it and the implementation in practical cases.

INDEX

CHAPTER 1. INTRODUCTION

We can catalogue the human recognition capacity as an extraordinary ability. Humans are able to store in our brain thousands of faces since we were babies and recognize other humans despite the years and distractions like glasses, beard etc.

For us, computational models of face recognition are interesting because of its applications. They can be applied into variety of problems including criminal identification, security systems or image and film processing. With a computational model properly trained it is possible to extract information of images, detect the face, normalize it and compare it with a database. Therefore, this model could be useful for the identification and recognition of people searched by police for instance.

The aim of this thesis is the implementation of a watchlist of characters built from pictures taken from internet able to be tested by a recognizer system for training (learn and store faces) and therefore recognize unknown images. The watchlist created will be a set of pictures properly processed having quite the same properties from size, view and eyes alignment point of view to train the recognizer module and be able to reach a high success rate of recognition. Therefore, to achieve that, 3 contributions (also detailed in **1.2** section) has been implemented:

- Creation of a watchlist from a database with pictures taken from internet sources (Google search, Facebook, Twitter...) to be used to train the system proposed. Two ways of watchlist creation have been proposed: Manually and Automatically. Both watchlists will be tested by the recognizer.
- Implementation of a normaliser module able to adapt pictures from internet sources into a watchlist.
- Implementation of detection algorithms based on Adaboost classifiers able to detect faces and eyes in photos to create a watchlist automatically.

The result of our contributions are two training set created manually and automatically and be compared its efficiency with the recognizer.
For both ways, the results have been successful (the system works) but with a low recognition rate (the results and performance will be evaluated on Chapter 4).

1.1. State of the Art

Face detection

Nowadays face detection functions are used on several applications such as photo cameras face identification, videogames, marketing or on the subject we are interested right now: Face recognition.

However we need to differentiate those two concepts. We need to have clear that face detection is a computer technology that determines the locations and sizes of human faces in arbitrary digital images. It detects facial features and ignores anything else such as bodies, objects or landscapes [2].

Face detection is the most important step for face Recognition because before recognizing a person by his face we need to know where the face is. This is important to discriminate all the rest of information from a picture to delete the irrelevant information for a computer to recognize people.

The face detector used on this thesis has been initially proposed by Paul Viola [3] and it is based on using classifiers named as "Haar classifiers" (based on Haar function proposed by Alfred Haar on 1910 [4]). Those classifiers are trained with the image we want to detect (called positive examples). Those positives examples are scaled into the same size (20x20 pixels). Then, negative arbitrary images of the same size are also introduced. To achieve on having a good classifier, this training should be done with a thousand of positive examples.

After the classifier is trained, it should be applied on a picture taking into account the region of interest. If the evaluation of this region matches with the object we want to detect the classifier will output "1" and "0" otherwise. Therefore it is possible to apply the classifier in the hole image moving the region window. The region window can be also easily resized to find objects with different size and doing the scan several times.

For this thesis we have used several Haar classifiers proposed by OpenCV environment trained with faces and eyes.

As we have said before, the classifiers used on this thesis are named "cascade boosted classifiers working with haar-like features". Meaning that the main classifier is built by different weak ones. So, with "cascade" we are referring that this weak classifiers are applied on several stages modifying parameters such as the size of the region of interest to detect objects at different sizes. “boosted” means that each classifier is implemented due to find a general rule taking into account different weak classifiers rules. This general rule should include as most positive cases as possible and discard the wrong ones. In [5] it is explained how the combination of weak classifiers became in a strong one.

The first practical boosting algorithm, and currently used and studied in applications in numerous fields is called AdaBoost. The classifiers used on this thesis are based on AdaBoost system [6]. Adaboost is an algorithm proposed in 1995 by Yoav Freund and Robert Shapire explaining a method for joining weak classifiers to becoming an strong one [7].

Finally, regarding "Haar-like feature" concept, are the input to the basic classifiers. Meaning the region of interest that should be analyzed by the classifier, compare it with the threshold learned and determine if that region contains (or not) the desired object.

A part from detection methods based on classifiers, different techniques have been developed to find faces in a picture most of them based on skin detection. The main advantage of skin detection methods is that is a fast procedure to detect faces. However the skin colour depends on the illumination of the skin so it is quite easy on having different skin tones for the same person under different lights. Then, sometimes it is quite difficult to filter the image with the correct colour to take into account all type of skin tones.

The main idea of using this method is finding a good colour representation space able to discriminate all the background in pictures except the skin ones. So, as the majority of normal cameras has only RGB representation it is needed to implement other colour representation by software. The commonly colour space used for that is called HSV (Hue-Saturation-Value). This is the most used because the post filtering of the colour we want (skin colour) is easier (however it varies depending on race).

On figure **1.1** it is shown a simple skin filtering using HSV colour space. On figure **1.2** the same idea is shown taking into account that on this case colour red is filtered so the result will be the detection of red objects. All of those examples are taking from live cam.

When all the image is filtered and the skin part is shown the idea is try to locate the face on the it. To do that some functions given by OpenCV can be used. This functions are based on finding geometrical figures in a image and they are able to find squares, circles, etc. with a determinate size. So, taking into account the face form a function able to locate squares could be implemented. However it is needed to have care with the minimum size considered as a valid face, if not, any square detected on the image will be considered (wrongly) as a face if this square is filtered as the skin colour.

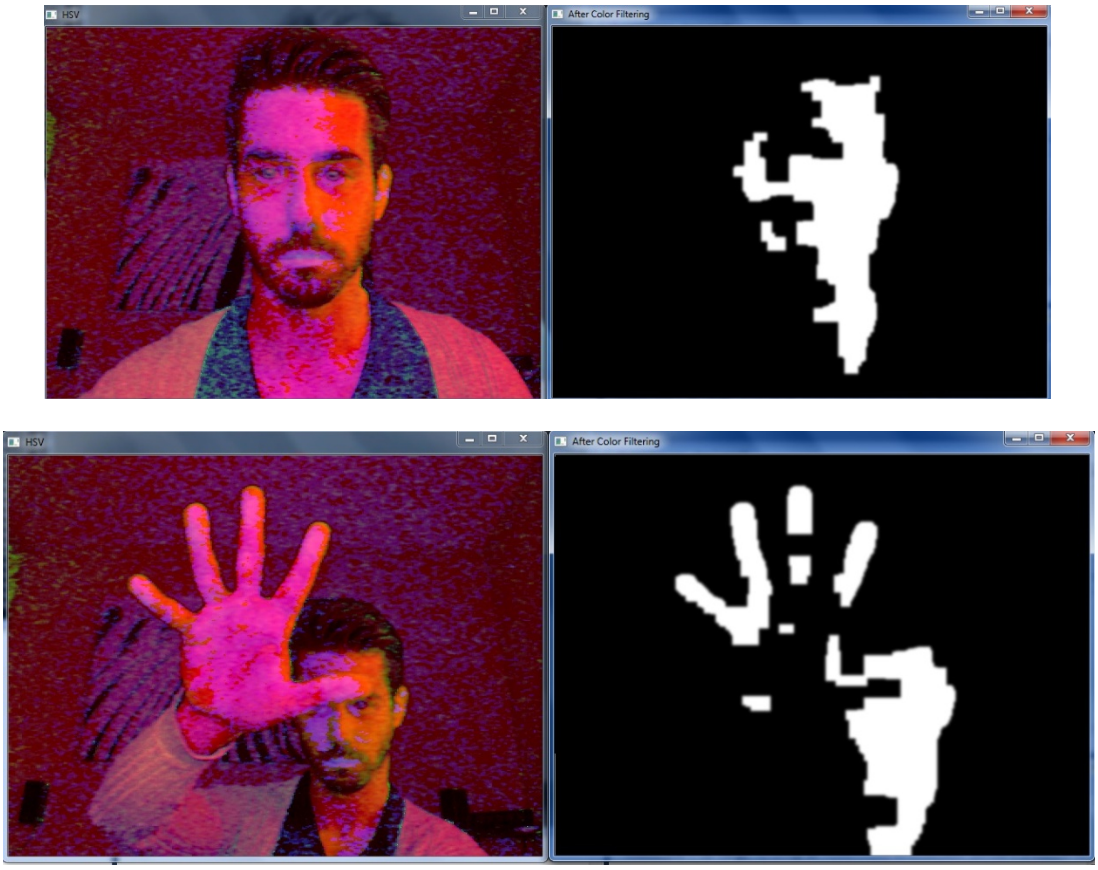

Fig. 1.1 Image example of HSV space conversion and skin filtering on live cam.

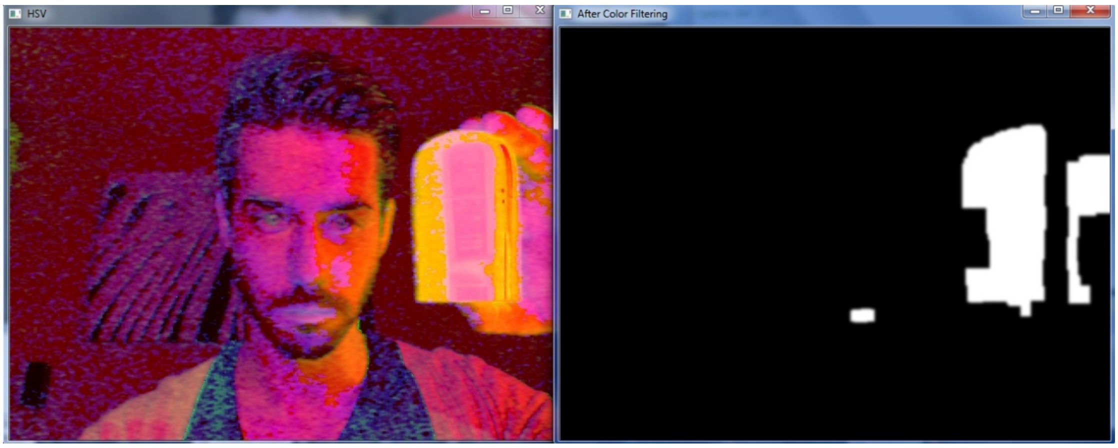

Fig. 1.2 Image example of HSI space conversion and red colour filtering on live cam.

The big drawback on using skin detection is that not all cameras have the same output RGB colour. Then the HSV will not be the same and the colour ranges for filtering can vary depending on the camera. Then, as mentioned those methods depends on illumination of the skin and it is not easy to find a correct colour filtering to include all the existing skins tones.

However it could be a good tool to recognize side faces, because the classifiers applied on this thesis are not good enough to detect them. So it means we will centre the study taking into account frontal faces pictures. However, as it is described in chapter 3 an skin detector will be used on this thesis in combination of Haar classifiers.

Face recognition

There are many ways that humans can identify each other, and so is for machines. Nowadays there are many different identification technologies available like Password/PIN (known as Personal Identification Number systems). The big drawback is that this techniques are not unique. It is possible for somebody to use the identification system by other person using its secret PIN number for instance.

There are other systems based on biometric identification like fingerprints or retina and iris recognition. However those techniques are not easy to use because the person needs to be positioned on a specific way and need to pause for a second due to be detected and therefore recognized.

Then, is face recognition able to give us more freedom deleting the necessity to stay right positioned to be recognized? And, if it is, which features do we need to recognize a face? It was shown by David Hubel [8] and Torsten Wiesel [9], that our brain has some cells that responds to specific local features like lines, angles or movement. Then the brain combine all this features into useful patterns to recognize daily things (persons, places...). So, face recognition is quite the same, consisting of extracting the important information from an image, putting them into a useful representation (like creating an image space representation) and doing a classification of them taking into account other representations from the same object on the same space.

Takeo Kanade describe on a PhD Thesis [10] a face recognition model based on geometric features meaning a way for recognition based on marked points on eyes, ears, nose... to create a vector and calculating the distance and angle between other reference vectors to confirm if it was or not a correct face recognition. This method is robust against illumination but it is needed an accurate method to select where eyes, nose, mouth etc. are.

Therefore, Brunelli, R proposed another recognition work based also on geometric features [11] but more complex. The idea is quite the same than Kanade but taking into account vectors of 22 values not containing only the geometrical features of the characteristics points. However, to manage this huge information requires a lot of processing.

For that reason, another ways of representing the important information of a face was thought. For instance like the eigenfaces method (used on this thesis).

This method was described for recognition by Turk, M. [12]. The idea is to describe a space where represent the important features of a face. This space is commonly named as "face space" and it is found with Principal Component Analysis (PCA). Applying PCA (shown in chapter 2) into an image we are able to represent it with a point on this space. If a new image income into the system its representation can be compared with the rest (previously represented) and if this new point is close enough into the rest we can assure that this new image corresponds (or not) with a face learned.

To achieve a recognition for different condition images it is needed to train with pictures with several characteristics (frontal view, 45 degrees view…). However, our approach consists on training and recognizing faces always in the same view. To achieve that, two systems of image processing are being implemented. They both will try to centre the face, align the eyes, put both eyes on the same distance and then cut the image with an specific pixel X and Y size. The difference between both systems will be the way how this operation is done. The first one is a manual operation and the user selects the position of the eyes on the image, and then the image is processed. The second one is an automatic operation and the system detects the face and the eyes on the picture by using face and eyes detection systems based on Adaboost classifiers. Once the eyes positions are set, the image is processed.

Both systems will be used to create a watchlist able to be tested the recognizer module system (training it) and recognize new characters.

1.2. Motivation and contributions

The goal of our thesis is the creation of a watchlist able to be tested by a recognizer through the training task. once trained, this recognizer should be able to recognize characters from an unknown input image.

In order to achieve our goal we have focused our work on giving to the recognizer module the proper data (watchlist) in order to make easier the task of recognizing. That proper data consists on faces pictures of known characters properly processed. The system functions by projecting faces images onto a feature space that includes the significant variations among known faces. The significant features are known as "eigenfaces" and they are the eigenvectors (principal components) of the set of faces.

The idea of the operation is to characterize an individual face by a weighted sum of eigenfaces. And then, compare their weights with the known weights faces.

In our approach the system has the ability to learn using a set of known pictures. This action is known as training. Therefore, the system is able to recognize new faces from pictures extracted from internet sources. This is a challenge since arbitrary internet pictures have many different features between them such as size, angle, zoom, colours, face point of view, and different backgrounds, which contain many information (landscapes, objects, etc.) not useful to recognize the character.

So, the contributions of this thesis in order to design an efficient recognition system according to the goals established are as follows.

The first contribution is the creation of a watchlist from an images database obtained from internet sources of many characters. The main idea is to train the recognizer with this watchlist in order to recognize as many characters as possible. This watchlist, as mentioned before, is created from pictures properly processed from characters obtained from Google Search, Facebook, Twitter, etc. So it means that doing the proper image processing (second contribution) it is possible to use pictures with different features and try to normalise them.

Therefore, the second contribution of this thesis is to create a normaliser module able to adapt the images from internet sources into a watchlist. To do that we have created a system image processing able to discriminate all the picture except faces and eyes (necessary to recognize). Normaliser module selects (detects) the faces and eyes on the pictures and scale them in order to have all the images at the same size, aligned at the same height (taking into account the eyes) and having the same eyes distance in all of them.

Finally, related with the third contribution of this thesis it has been necessary to implement a simple face and eyes detector using classifiers based on AdaBoost. The main aim has been to complement those classifiers to make the detector as robust as possible meaning implementing different faces and eyes classifiers in cascade, implement a skin detector or implementing a function to discard multiple eyes detections and use the best detection taking into account the overlap of them using matrix overlapping. Then, tasks has been implemented in order to detect more faces and eyes on pictures consisting on turning left/right the image if we don’t detect any face or eyes on the first stage.

The next chapter will include the watchlist used for this thesis, it means the training set used to recognize the new image inputs and how it is built taking into account the problems we have on using pictures from internet sources.

1.3. Thesis content

This section will summarise the content of the thesis for the next chapters.

Chapter 2. Internet database watchlist application explains the watchlist used to train the recognizer system, the methodology followed for its creation and how we achieve recognizing characters. The watchlist is obtained from a database formed by pictures taken from internet sources. Those pictures, properly processed will be used for training the system. We have collected 10 characters with an average of 70 pictures each one (the number of pictures vary depending on the character). Each picture has been processed to obtain an standard picture (normalized). Then the idea is to train the system with the characters (normalized pictures) due to provide to the system the ability to recognize an input unknown picture. To test the overall system, 2 different experimental set ups were considered. They are specified below:

- 1 training with 6 watchlist characters.
- 1 training with 10 watchlist characters.

As it is explained above, it is important to have a good watchlist to learn. This watchlist should be created to train the system before starting the recognition process. The watchlist is built by using the normalized images obtained from database (different random pictures from Internet sources), so it is needed an additional system to normalize them. In fact, two different ways were implemented to generate this watchlist: automatically and manually, which are both explained in detail in chapter 3. So, it means that the 2 scenarios described above become in 4 taking into account that they will be applied with those two watchlists.

Finally a general module view of the system is presented from the database (internet pictures) till recognizing characters from unknown pictures.

Chapter 3. Detection and recognition system implementation includes the following sections: OpenCV, Face database for raw data, preparing the data: normaliser stage, manually watchlist creation, automatic watchlist creation, training, and recognition and classification. OpenCV [1] (Open Source Computer Vision Library) libraries and C++ language have been the programming environment used for the development system. Face database for raw data section describes the necessity on having an accurate normalized database (watchlist). Preparing the data: normaliser stage section detail the module that process all the images to adapt them with the same features. Manually and automatic watchlist creation describes how both watchlists have been built. Training section explains how the data of training set is given to the recognizer. Finally the recognition and classification section describes which is the returned value of the system. In this chapter is also explained some code variables to have an brief idea of how the detection part works.

Chapter 4. Results and discussion includes the detailed description of the 4 experimental scenarios and their implementation. The results obtained in each case are also discussed.

Chapter 5. Conclusions and future work shows the achievements of this thesis and new ideas as future works with the aim of improving the results obtained. As the system has been implemented on a modular way we can for instance change the parameters used to create a different featured watchlist able to train and be tested by the recognizer module. Finally, several possible related works with this thesis subject have been proposed.

CHAPTER 2. INTERNET DATABASE WATCHLIST APPLICATION

The general idea of face recognition and we want to achieve is extract the relevant information in a face image, encode it and compare one face encoded with a watchlist of faces encoded in a similar way.

Therefore, as mentioned before, the recognition method used on this thesis is based on "eigenfaces" approach. Basically it consists on extracting the relevant information contained on an image of a face and capturing the variation in a collection of face images (watchlist), this information will be stored and will be used to be compared with unknown pictures or images entered as an input on the system.

So, in a technical language, what is tried to find is the principal components of the distribution faces, or what is the same, the eigenvectors of the covariance matrix of the set of face images, taking into account that the image is a point in a dimensional space. The eigenvectors are sorted, according the variation (weight distance) among them.

These eigenvectors can be thought as a set of features that together characterize the variation between face images and each individual face can be represented in terms of a linear combination of the eigenfaces.

The best M eigenfaces (it means the with largest eigenvalues, so implies having more information because the variance is higher) forms the "face space" of all possible images to create the watchlist and training the recognizer.

The idea of using eigenfaces was motivated for the efficiency representation pictures of faces using principal component analysis (PCA).

The purpose of PCA is to compress the 2-D data obtained from an image to be represented as 1-D vector. However a strong correlation between training images from the same character should be exist to have their 1-D vector representation as much closer as possible to create face classes of each character inside the face space. So this is why it is needed first to normalize all the images that will be used for training.

Therefore, each face in the training set is transformed into the face space and its components are stored in memory. What it means that the face space will be formed with the known faces.

When an input face is given to the system, it is projected onto the face space. Then, system computes its distance from all the stored faces.

Once this operation is done, two cases will be taken into account regarding the result of distance computation:

Known face: If the result is near to a stored face.
Unknown face: If the result is not near to a stored face.

So, as we have seen, depending on the case the system can classify the face in known or unknown face.

As mentioned, the normalization task is needed to achieve strong correlation between the images that will form our watchlist and have the face classes properly delimited taking into account that they need having the same (as much as possible) view features. To achieve that, the following requisitions have been imposed:

- Align the eyes in the same Y axis.

- Put the eyes at the same distance in all pictures making zoom out/zoom in at 50 pixels. This variable can be modified.

- Select the face part of the images once eyes have been detected or marked by the user aligning the eyes on X axis.

- Cut the images at the same size at 160x200 pixels.

- Put right/left and/or up/bottom bars if the image doesn't have enough pixels to be cut at 160x200 pixels.

- Convert the image to black and white to reduce the noise produced by colour.

Then, once the pictures are properly normalized the recognizer module can start its task. The approach used on this thesis for face recognition involves the following operations. From one hand the recognizer should be trained before starting recognizing and this operation include:

Step 1. Acquire initial set of faces images (training set) by the watchlists that have been built taking into account manual and automatic methods.

Step 2. Calculate the eigenfaces from the training set, keeping only the M eigenfaces that correspond to the highest values. Those M images as it has been said before will define the face space.

Step 3. Calculate the corresponding distribution in M-dimensional weight space for each known individual by projecting their face image onto the face space.

After, once the system is trained the following operations will be done to recognize new images:

Step 4. Calculate a set of weights based on the input image and the M eigenfaces by projecting the input image onto each of the eigenfaces.

Step 5. Determine if the image is a known or unknown face by checking to see if the image is close enough to any face class taking into account the projections of the training set in the face space.

Taking into account the mentioned steps a general view in figure 2.1 of the system will be shown.

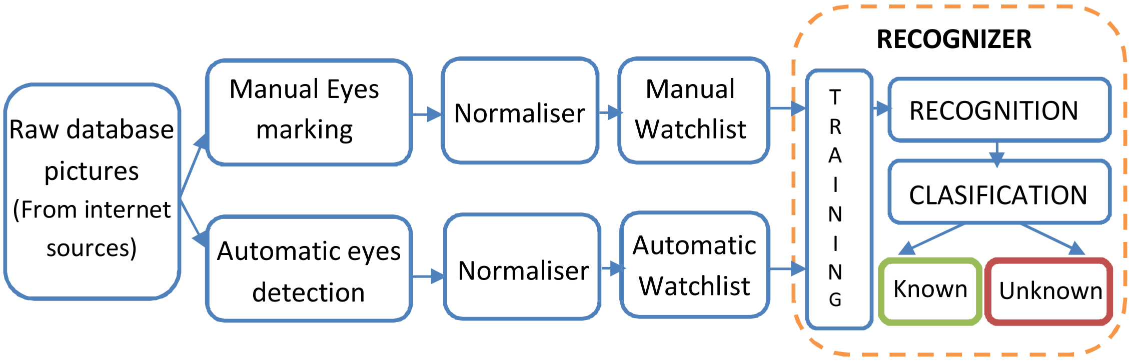

Fig. 2.1 General scheme view of implemented recognition system

As we have said before, raw database pictures will be obtained from any source in internet that means we will obtain different views, angles, sizes etc. of the same character. So, the importance of normaliser is very high to obtain a good normalized image able to obtain an accurate and well delimited face class for each character.

The manual eyes marking will be done by the user. The coordinates will be stored in a file to be used in the future. Doing that we don't need to mark again all the pictures if we need to train the system again for instance with different values of image size, eyes distance, etc.

The automatic eyes detection will be done using Haar classifiers in cascade. To detect all of them with the higher detection successful rate it have been implemented 4 classifiers (one for normal front eyes view, one for front eyes behind glasses, one for right eyes and one for left eyes) in cascade. However the first approach is detect the face and then detect the eyes inside the face part given in the previous stage. To detect the face 3 Haar classifiers in cascade will be implemented.

To measure the recall and precision parameters of those classifiers 6 characters has been taken into account. With recall parameter we can see how many good detections have had the classifier from all desired detections. With precision parameter we can see from all the detections done which ones are right. The results are quite acceptable and shown in table 2.1 and table 2.2.

Both parameters (recall and precision) are calculated after the 4 eyes filters put in cascade and after the 3 face filters put in cascade.

	Good detections	Total detectable elements	Recall
Will Smith Right Eyes	54	56	**0,96**
Will Smith Left Eyes	54	56	**0,96**
Will Smith Faces	52	56	**0,92**
Jessica Alba Right Eyes	76	76	**1**
Jessica Alba Left Eyes	76	76	**1**
Jessica Alba Faces	76	76	**1**
David Beckham Right Eyes	50	52	**0,96**
David Beckham Left Eyes	51	52	**0,98**
David Beckham Faces	52	52	**1**
Johnny Depp Right Eyes	60	70	**0,85**
Johnny Depp Left Eyes	61	70	**0,87**
Johnny Depp Faces	59	70	**0,84**
Natalie Portman Right Eyes	76	77	**0,98**
Natalie Portman Left Eyes	76	77	**0,98**
Natalie Portman Faces	77	77	**1**
Jennifer Aniston Right Eyes	74	76	**0,97**
Jennifer Aniston Left Eyes	71	76	**0,93**
Jennifer Aniston Faces	74	76	**0,97**

Table 2.1 Recall calculation for 6 characters (eyes and face filters).

	Good detections	All detections	Precision
Will Smith Right Eyes	54	56	**0,96**
Will Smith Left Eyes	54	54	**1**
Will Smith Faces	54	55	**0,98**
Jessica Alba Right Eyes	76	76	**1**
Jessica Alba Left Eyes	76	76	**1**
Jessica Alba Faces	76	78	**0,97**
David Beckham Right Eyes	50	52	**0,96**
David Beckham Left Eyes	51	51	**1**
David Beckham Faces	52	52	**1**
Johnny Depp Right Eyes	60	63	**0,95**
Johnny Depp Left Eyes	61	65	**0,93**
Johnny Depp Faces	59	71	**0,83**
Natalie Portman Right Eyes	76	76	**1**
Natalie Portman Left Eyes	76	76	**1**
Natalie Portman Faces	77	77	**1**
Jennifer Aniston Right Eyes	74	75	**0,98**
Jennifer Aniston Left Eyes	71	72	**0,98**
Jennifer Aniston Faces	74	76	**0,97**

Table 2.2 Precision calculation for 6 characters (eyes and face filters).

The results show that from one hand we are detecting almost all the elements we want and from the other hand show that we are not doing many faults on detecting.

Both (manual and automatic) eyes marking systems will be explained in detail in chapter 3.

The normaliser module will have the same operations described before for both eyes marking systems. However the implementation is quite different because the origin of the eyes coordinate. On automatic way the coordinates are directly passed from detection module to normaliser one. However in manual way the normaliser get the coordinates from a file were have been stored.

The Recognizer module can be divided in three differentiated parts: The Training module, recognition module and classification module. The training module will create the face space with all the pictures of different characters we will introduce as its input for training, as first approach we will train with the entire watchlist and then as a second approach with 20 pictures of each character.

The system is implemented to train with all the pictures we insert except the last one that will be used to be recognized (we will see in chapter 3 how the training set and the image to recognize is introduced). This last picture will be processed by the recognizer and it will calculate its weights and compare with the weights of watchlist pictures. Finally it will decide if the input image is close enough from any of the face classes stored. If it is it will be classified as known. If it doesn't, it will be classified as unknown.

So, one aim is to compare the recognition effectiveness of the recognizer using both ways of creating our database (manual and automatically) and see which database is better to use from the reliability point of view. However what is more important with this comparison is seeing if the automatic way of watchlist creation works effectively and see how wrong is compared with the manual one.

On chapter 3 we will see how the system works from a technical point of view.

CHAPTER 3. DETECTION & RECOGNITION SYSTEM IMPLEMENTATION

In this chapter it is shown the way how the system works from a technical point of view. It means it is shown and explained some important code parts to understand the operation and which is the programming environment used to implement the system.
One requirement set on this thesis has been to use OpenCV libraries as a software to use for recognizing and detecting tasks that it offers. Then, to interact with those libraries C++ language has been used.

3.1. OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.
OpenCV is released under a BSD license (a family of permissive free software licenses, imposing minimal restrictions on the redistribution of covered software) and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. [1]

From the 2500 optimized detection algorithms that OpenCV provides we will focus for this thesis on face and eyes detection ones.

In OpenCV 2.4 (used on this thesis) the currently available recognition methods are:

- Eigenfaces (Based on PCA classifier). Used in this thesis.
- Fisherfaces (Based on Linear Discriminate Analysis classifier). Not used on this thesis. See [13] to see the differences between Eigenfaces and Fisherfaces methods.
- Local Binary Patterns Histograms (Based on local binary patterns: 1 and 0 to represent the data). Not used on this thesis. See [14] for Recognition method using Local Binary Patterns).

We will use OpenCV for implementing detection methods. OpenCV uses Haar classifiers and the data is stored in an XML.

As mentioned, in this thesis a method based on Eigenfaces will be implemented. It means we will use only the XML files based on this method.

3.2. Face watchlist for Raw data

As we have mentioned on previous chapters, to starting on recognizing faces we need first to have a watchlist of the characters needed to train the system. However we need to have an accurate watchlist easy to be used by the classifier. When we talk about "easy" concept it means to have a watchlist of different images with quite the same characteristics between them to train the classifier. If the classifier is properly trained with properly images then the recognition of one external unknown image will be more effective. To do so, we need to prepare the data before training the classifier. Then the main idea is having as many pictures in watchlist as possible from different characters to have as much successful recognition rate as possible.

3.3. Preparing the data: Normaliser stage

To prepare the data we need to set first some parameters to be taken into account for normalising and be applied in all the images that will form our watchlist. PCA system will work better if all the input information have the same characteristics so, the normalise module described in chapter 2 has been implemented taking into account some specific requirements. The aim is to use the eyes as a reference to normalise the pictures having the same features and create our watchlist, discriminating the rest of the picture that doesn't contain any face.

The normaliser stage will apply the same procedure for manual and automatic watchlist creation way, the difference as mentioned is the way for obtaining the eyes position. So, as we need the eyes as a reference to normalise, on the first case, the location of the eyes will be marked manually and on the second case the location eyes will be marked automatically using eyes detection methods. Once done, the normalise module will do the following operations:

- **Eyes align**: To do so, we will turn right or left the image to put them aligned.

- **Eye distance**: We have decided to create the watchlist with images having always the same distance from both eyes. The intention is not to deform the images if they have not the proper size (to do so it is needed to zoom-in/zoom-out).

- **Centred eyes**: We will centre the eyes on the image to know exactly where the eyes are. We will centre them on X and Y axis.

- **Size of the images**: To train the recognizer each image will be processed so it means that we are interested on images with little weight, for that reason we have decided that too not lose too much information the images will be 200x160 pixels. So, when the image will be managed taking into account the previous parameters we will select the mentioned size.

- **Left/Right and Top/Bottom bars**: Finally if the portion image selected doesn't reach 200x160 pixels we will fill it adding the proper black bars to maintain the position of the eyes.

- **Colour**: The image will be passed from RGB to black and white.

However, before normalise the data for the watchlists, we need to give to normalise module each picture with the eyes information. To detect them it has been implemented to ways to do it: Manually and automatically.

3.4. Manually watchlist creation

As we have said, before starting recognizing faces we need to prepare the data due to make the recognition task easier for the recognizer and be able to do it as effective as possible.

For manually database creation mode, the main idea is to mark (manually by the user) where the eyes of each picture are and then create an standard image from each one (normalize it).

Then, the first implemented thing is a function that allow to user mark (with mouse cursor) where the centre (pupil) of both eyes are on each image. This information (for each image) is stored into a file due to be recovered in the future if needed.

Once both eyes centre are marked, the next function rotates the image (at left or right) to align the eyes on the same X axis.

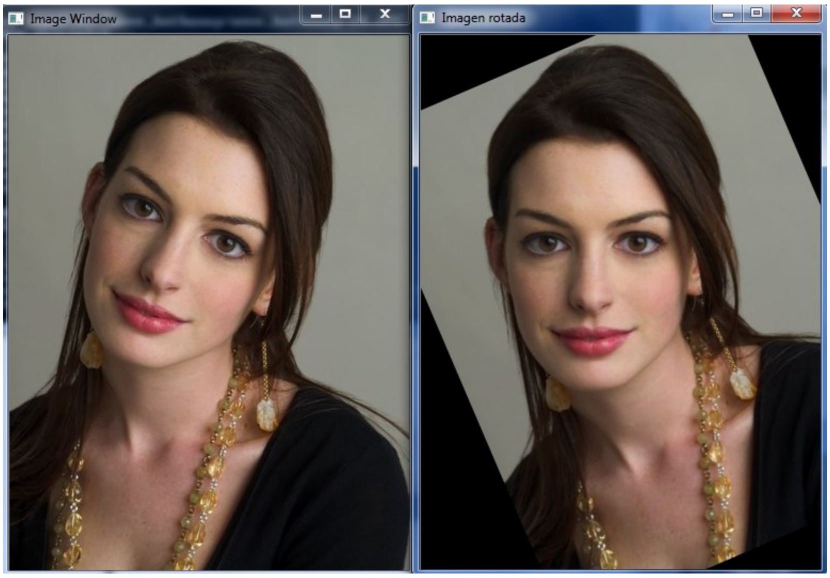

Fig. 3.1 Before and after image rotation for eyes alignment.

The next function will zoom in or zoom out the image taking into account that the eyes have to be separated a specific distance. For instance, if the user eyes marking has 100 pixels of separation and the eyes distance is set to 50 pixels the image will be zoomed in until reach the correct separation. What it means to zoom out the image 2x.

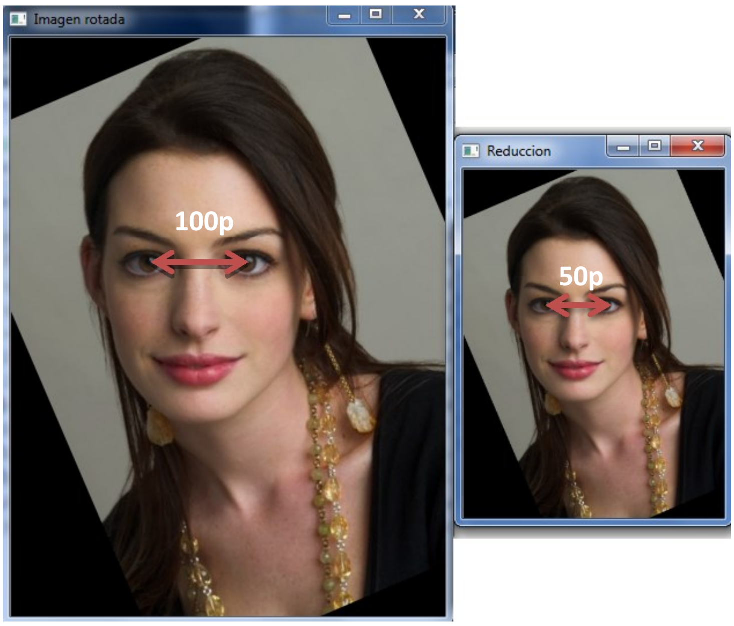

Fig. 3.2 Before and after scaled image for 50 pixels distance between eyes example.

Then, when the image is correctly zoomed it is needed to cut it with a proper size. However we need to do this operation taking into account that the eyes must be centred on X and Y axis. In the X axis the eyes will be in the middle of the image and in the Y axis the eyes will be at 2/5 in the image. It means that if the image is 200x160 and the eyes centre are separated 50 pixels they will have 55 pixels each side on X axis and 80 pixels from upside and 120 pixels from bottom on the Y axis.

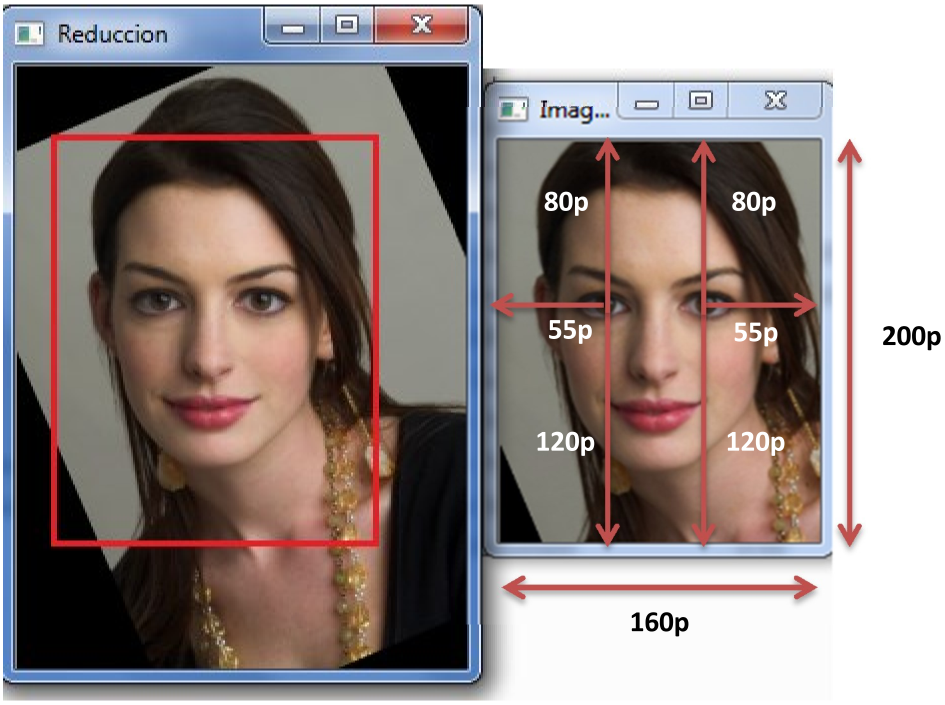

Fig. 3.3 Proportion size of image for 50 pixels distance between eyes example before and after the cut.

If we try to select the required pixels around the marked eyes points and we don't have enough pixels to have an image of 160x200 pixels it is needed to fill it with black ones. It means adding black bars on the left/right or up/bottom sides.

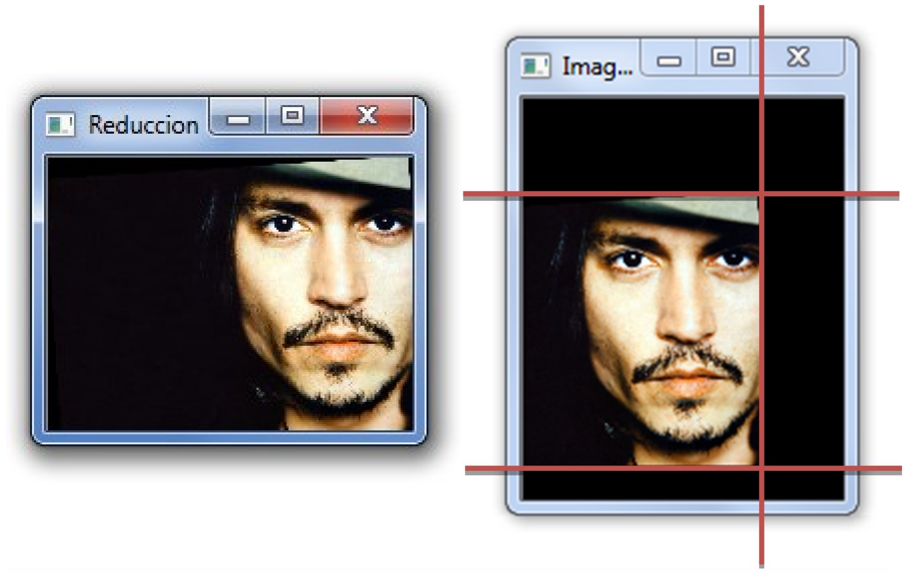

Fig. 3.4 Before and after adding bars on image for 50 pixels distance between eyes example.

Finally the last step for normaliser module is change the colour of the image from RGB to black and white.

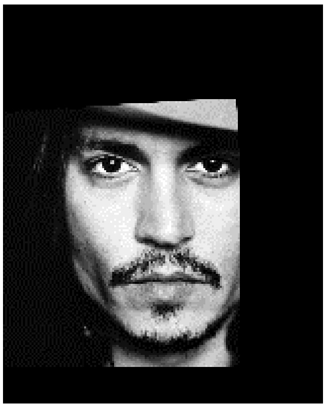

Fig. 3.5 Conversion to Black and white and final result image to store in database for 50 pixels distance between eyes example.

3.5. Automatic watchlist creation

For automatic watchlist creation the main idea is having a program that detects automatically where the centre of the eyes of each image are. To do so it is needed to implement a function able to detect where the eyes are and then calculate the centre.

The procedure has been thought to be as efficient as possible. It means we have implemented a previous stage where we detect the faces (or faces) in the image and then we only pass the face detection result to eyes classifiers.

Doing this, the eyes detector will always receive as input an image with a face where the eyes classifiers have to detect both eyes. After detecting the eyes and generate an square around them the pupil will be calculated as a result of calculating the centre of this square. Then the idea is to see if this approach is useful for recognizer once the database will be created.

3.5.1. Face detection

As mentioned, for detection, Haar cascade classifiers have been used. Those classifiers are .xml and they will be loaded and they will be applied on each image we want to detect. For image detection we have used the following ones:

- haarcascade_frontalface_alt.xml
- haarcascade_frontalface_alt2.xml
- haarcascade_profileface.xml

Those 3 files should be previously loaded (with `cvLoad()` function). It means we need to open them and storage it into a variable (named as `face_cascade`, `face_cascade2` and `face_cascade3`).

Then each image will be passed through the following function:

```
CvSeq * pFaceRectSeq;

pFaceRectSeq = cvHaarDetectObjects
                (frame, face_cascade, pStorageface,
                1.1
                3,
                CV_HAAR_DO_CANNY_PRUNING,
                cvSize(40,40));
```

On structure `pFaceRectSeq` (`CvSeq` type) we will store the number of faces detected, the size of each one and other parameters that we are not interested in right now.

As it has been indicated before, for face detection it has been used 3 haar classifiers (see Annex B for detection results using 1 or 3 classifiers) so it means that if the first one don't detect any face we will apply the second one (`haarcascade_frontalface_alt2.xml`) and if the second one don't detect anything as well a third one face classifier (`haarcascade_profileface.xml`) will be applied.

Taking into account that those classifiers can fail it has been implemented an skin detector (developed by the other thesis director Raúl Quijada) able to discriminate all the information in the image that is not significant to detect faces. Note that this function is applied if the previous 3 face classifiers fails applied directly on the input image.

This skin detector will be applied on two cases: If we have not detected any faces with previous 3 classifiers or if the detection results in more than one face in the image. For creating our watchlist we are interested in using images where only one face appears but if we want to implement this detector for other purposes and use it with pictures with more than one face it will work without problems too. However this detector will helps us also to eliminate false detections as in figure 3.6 appears.

Fig. 3.6 False face detection in image. The false one will be discarded thanks to the skin detector.

The skin detector will erase (it will put in black) the image part that doesn't contains the skin colour configured in the function.

Once, the skin detector is applied and the picture is filtered, the 3 face classifiers will be applied again. It is important to remark that this face filter is not applied always before applying the 3 face classifiers because it doesn't work if the input image is black and white. If the input image is a grey scale image and we apply the skin detector all the image will became in black so the result will not be valid to be processed by any classifiers.

Fig. 3.7 Right eyes detection thanks to right face detection helped by skinDetector.

Arrived into this point, if the classifiers have not detected anything it has been implemented a function to rotate the image to left and right the image (25 degrees) and apply again the 3 face classifiers for each rotation.

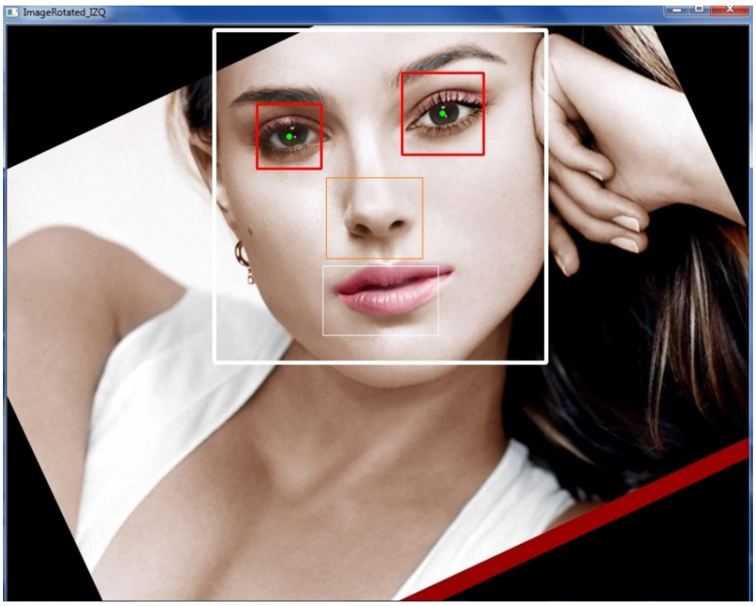

Fig. 3.8 Example of non face detection image until the it is rotated left 25 degrees.

Finally, the results (all faces finally detected) as we have seen in previous figure are shown and a white square is drawn around the face. This square portion of the picture will be send to next and most important stage: Eyes detection function.

3.5.2. Eyes detection

For eyes detection we have used the following .xml classifiers:

- haarcascade_eye_tree_eyeglasses.xml
- haarcascade_eye.xml
- haarcascade_mcs_lefteye.xml
- haarcascade_mcs_righteye.xml

The aim for eyes detection algorithm is obtaining the position of the eyes (pupil) as in the manual marking way. However we want to obtain it automatically. So, first we need to detect where the eyes are and then calculate its centre.
As we have said before, the idea is having only one face per image. However this code has been implemented due to detect all the eyes from all the faces we detect in the input picture.

To do that, and after loading eyes classifiers (as we did with face classifiers) we will apply them to each face by this function:

```
pStorageeye = cvCreateMemStorage(0);

pEyeRectSeq = cvHaarDetectObjects
                (copy_eyes, eye_cascade, pStorageeye,
                1.1,
                3,
                CV_HAAR_DO_CANNY_PRUNING,
                cvSize(5,5));
```

The detection result is an structure "pEyeRectSeq" that contains the coordinates of the bounding box detected for the eyes.

As we have 4 classifiers, we have implemented 4 stages in cascade to detect as much eyes as possible if previous stages don't detect the correct ones. Then, the final result should be the draw of one left eye and one right eye selected from all detected. To achieve that it has been implemented a function able to compare all the results taking into account the overlapping between all of them using matrix. If the overlapping is more than the 40% the conclusion is that both detections corresponds with the same eye (right or left) and we continue looking for more eyes and comparing with the rest of detection in pairs.

If after the 4 stages for detection (see Annex B for detection results using 1 or 4 classifiers) the classifiers don't have detected any eye or only one, the image will be rotated left and right and we will pass-through the image into face classifiers and then eyes classifiers as well. We do that because it is possible on having a good face detection but bad eyes detection because of the face is rotated. We can see an example of mentioned problem in figure **3.9**.

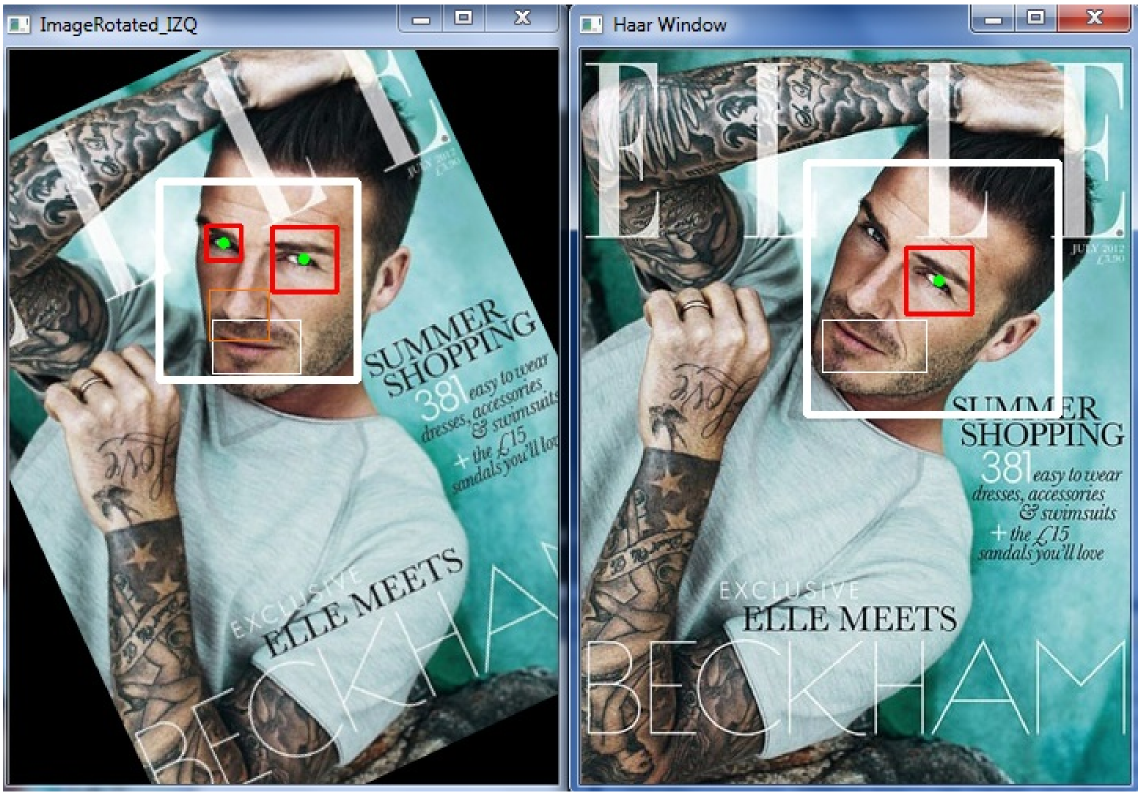

Fig. 3.9 Example of non both eyes detection image until the it is rotated left 25 degrees.

Once both eyes are considered as valid detection, then we calculate the centre of the bounding box (size of each eye detection) trying to reach the centre of the pupil by considering the middle of the bounding box in X and Y axis as the size of the bounding box is easily obtained from the structure result mentioned "`pEyeRectSeq`" (`CvSeq` type) that contains the number of detection results for the classifier applied and the size of each detection.

Finally, we will draw the results with a red square marking the eye detection and green point to represent the centre of the bounding box that we will used as a coordinates to be used by normaliser module to create a valid image for the watchlist taking into account the requirements established (size of the image, distance between eyes, etc.).

3.6. Training

The next step for the system is to train the recognizer. This is the previous step for recognize new images. The code used for this module is proposed by the programmer Philipp Wagner and we have use it to test the efficiency of our watchlists created by manually an automatically way.

The main idea is introducing as many images from the same character as possible to create face classes easy to be delimited. It means, as we have said before, we need to enter the images with similar features to have strong correlation between them.

To do that, we have used all the images obtained from normaliser module to create our watchlists. In our case, training, recognition and classification module are implemented jointly so we will introduce N images to train and the system will train with the image in the last position. The last one will be the image to be recognized and classified as known or unknown. To introduce the images to learn it has been created a .txt file with the information of the pictures as follows:

```
resources/watchlist/WSmith_img_0.JPG;1
resources/watchlist/WSmith_img_1.JPG;1
resources/watchlist/WSmith_img_2.JPG;1
[...]
resources/watchlist/NPortman_img_0.JPG;2
resources/watchlist/NPortman_img_1.JPG;2
resources/watchlist/NPortman_img_2.JPG;2
[...]
resources/watchlist/JDepp_img_0.JPG;3
resources/watchlist/JDepp_img_1.JPG;3
resources/watchlist/JDepp_img_2.JPG;3
[...]
resources/watchlist/DBeckham_img_48.JPG;6
resources/watchlist/DBeckham_img_49.JPG;6
resources/watchlist/DBeckham_img_50.JPG;6
resources/watchlist/DBeckham_img_51.JPG;6
```

In this file, it is shown the directory of the learning images and the identifier number of the character at last. It means that the result of recognizing and classification will be the ID of the character that corresponds with the recognition. The last image will not be used to train, it will be the input unknown image to be recognized

3.7. Recognition & Classification

This is the last module of the system. The function is to compare the unknown input image weights with the rest of learned images and classify it as know or unknown image.

As we have said, the result of this module will be the ID of the character.

In the next chapter we start on doing the tests. This chapter includes the 4 scenarios described and the results obtained.

CHAPTER 4. RESULTS AND DISCUSSION

This chapter includes the results and discussions of the tests proposed:

- 1 training with 6 watchlist characters. The objective of this set up is to recognize the unknown character out of the 6 characters learned from watchlist.
- 1 training with 10 watchlist characters. The objective is see if this scenario compared with the previous one has the same recognition successful rate for the 6 previous characters with the new ones.

As it has been created two watchlists (manual and automatically) each scenario will be applied on each one modality.

4.1. Manual watchlist

As mentioned before, the aim of this modality function is create a watchlist based on user manual eye marking. To test the watchlist result it has been considered 2 scenarios to train the system and therefore use this training for recognizing.

4.1.1. 1xPCA Training and recognition (6 characters)

This training is done by using 6 characters (from ID = 1 to 6) following the format saw in the .txt file shown in 3.6 section. Only one PCA is done, it means the system will train with the 6 characters at the same time. In that case the image positioned in the last line of the file:

```
resources/watchlist/DBeckham_img_51.JPG;6
```

Will be used as an input (the system will not train with this image) to be recognized. For each character it has been used the following pictures to train.

- Will Smith: 55 pictures (ID = 1)
- Natalie Portman: 75 pictures (ID = 2)
- Johnny Depp: 69 pictures (ID = 3)
- Jessica Alba: 75 pictures (ID = 4)
- Jennifer Aniston: 74 pictures (ID = 5)
- David Beckham: 51 pictures (ID = 6)

The processing time of the total operation (train + 1 recognition) is about 25 seconds. The result is a successful identification of the character (David Beckham). In figure 4.1 it is shown the result where the class (ID) of input image is 6 and the prediction of the recognizer is also ID 6 what it corresponds with David Beckham on this case.

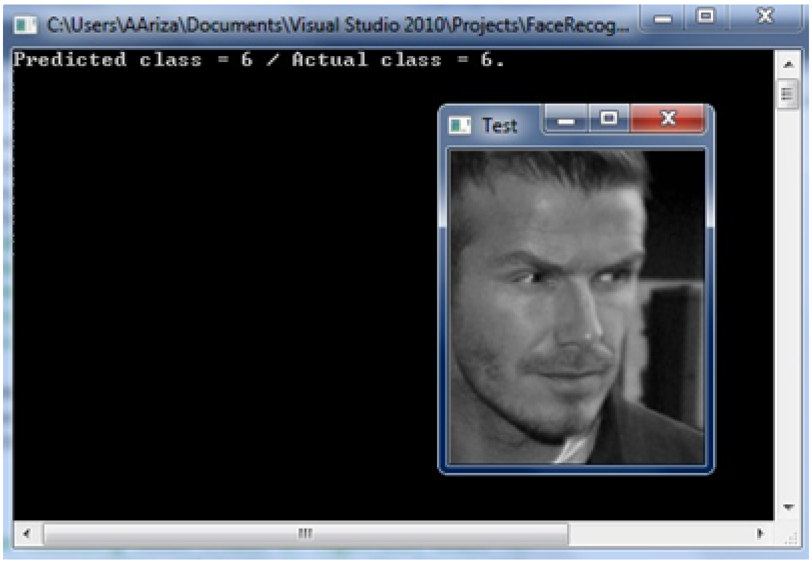

Fig. 4.1 Result for Face Recognition with 1xPCA manually.

Then, another training model and image detections inside this scenario.
For instance to train we have considered the pictures as follows:

- Will Smith: 20 pictures (ID = 1)
- Natalie Portman: 20 pictures (ID = 2)
- Johnny Depp: 20 pictures (ID = 3)
- Jessica Alba: 20 pictures (ID = 4)
- Jennifer Aniston: 20 pictures (ID = 5)
- David Beckham: 20 pictures (ID = 6)

And in this case it has been considered 10 different unknown pictures per character to recognize. The results are summarized on table 4.1.

	Pic 21	Pic 22	Pic 23	Pic 24	Pic 25	Pic 26	Pic 27	Pic 28	Pic 29	Pic 30
WSmith	Yes	Yes	Yes	No(6)	yes	Yes	No(6)	Yes	Yes	No(6)
NPortman	Yes	Yes	Yes	No(6)	No(4)	No(6)	Yes	Yes	No(4)	No(4)
JDepp	No(4)	No(6)	No(2)	No(5)	No(6)	Yes	Yes	No(2)	No(2)	No(4)
JAlba	Yes	No(2)	No(2)	No(6)	Yes	Yes	Yes	No(6)	No(2)	No(6)
JAniston	Yes	Yes	No(1)	Yes	No(2)	Yes	Yes	No(4)	No(4)	No(4)
DBeckham	Yes	Yes	Yes	Yes	No(2)	No(4)	Yes	Yes	Yes	Yes

Table 4.1 Summarized results for 20 pictures training and 10 unknown pictures input each character in manual wacthlist.

	% Recognition success
WSmith	70%
NPortman	50%
JDepp	20%
JAlba	40%
JAniston	50%
DBeckham	80%

Table 4.2 Summarized % recognition results for 20 pictures training and 10 unknown pictures input each character in manual watchlist.

The parenthesis number in table 4.1 indicates with which character the confusion by the recognizer has been produced.

It has been observed that for instance for Natalie Portman example in picture 24 from table 4.1 if we train with all the characters in the wacthlist but only with 10 images from David Beckham (Which it is mistaken with) the recognition became right. However if we reduce the training set of Natalie Portman into 10 the prediction became wrong again.

So, taking into account this results in this scenario it is needed to study the optimisation number of training pictures from the watchlist able to recognize as much images as possible.

4.1.2. 1xPCA Training and recognition (10 characters)

This training is done by using 10 characters (from ID = 1 to 10). Only one PCA is done, it means the system will train with the 10 characters at the same time. These are the number of pictures will be used for the watchlist for this training:

- Will Smith: 55 pictures (ID = 1).
- Natalie Portman: 75 pictures (ID = 2).
- Johnny Depp: 69 pictures (ID = 3).

- Jessica Alba: 75 pictures (ID = 4).
- Jennifer Aniston: 74 pictures (ID = 5).
- David Beckham: 51 pictures (ID = 6).
- Anne Hathaway: 74 pictures (ID = 7).
- Angelina Jolie: 82 (ID = 8).
- Al Pacino: 87 (ID = 9).
- Adam Sandler: 71 (ID = 10).

The processing time of the total operation (train + 1 recognition) is about 6 minutes. The result is not a successful identification of the character. In figure 4.2 it is shown the result where the class (ID) of input image is 10 (Adam Sandler) and the prediction of the recognizer is also ID 4 (Jessica Alba).

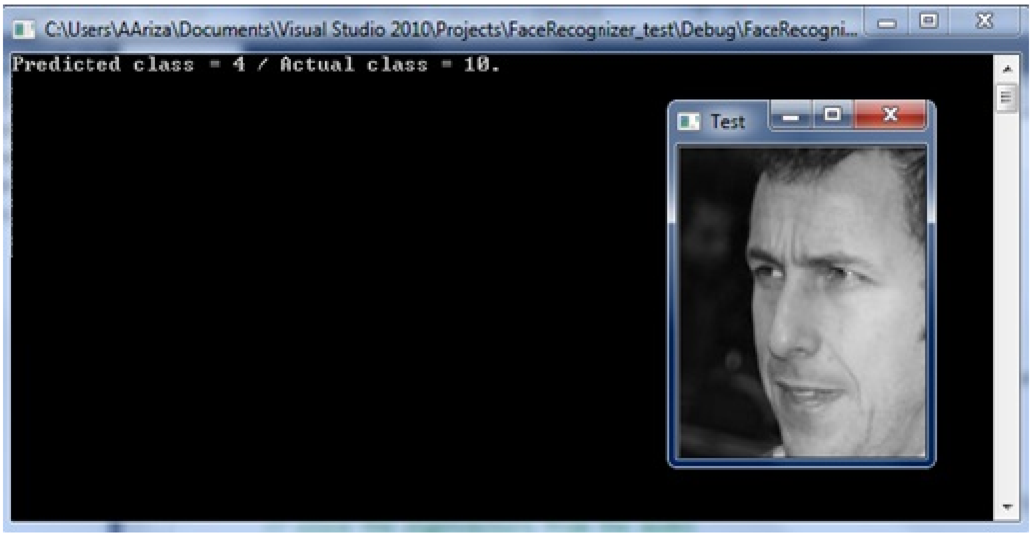

Fig. 4.2 Result for wrong Face Recognition with 1xPCA (10 characters) manually.

The result of figure 4.2 is done with the entire watchlist. So, to minimize times we will train with 20 pictures of each character and we will try to recognize 10 pictures from each one. Table 4.3 will summarize the results of this training and in 4.4 table the detection percentage will be shown.

	Pic 21	Pic 22	Pic 23	Pic 24	Pic 25	Pic 26	Pic 27	Pic 28	Pic 29	Pic 30
WSmith	No(10)	No(9)	Yes	No(6)	Yes	Yes	No(6)	Yes	No(10)	No(6)
NPortman	No(10)	No(9)	No(9)	No(7)	No(4)	No(10)	Yes	Yes	No(8)	No(4)
JDepp	No(4)	No(6)	No(2)	No(10)	No(8)	Yes	Yes	No(2)	No(2)	No(4)
JAlba	Yes	No(2)	No(2)	No(9)	Yes	Yes	Yes	No(10)	No(2)	No(8)
JAniston	No(8)	Yes	No(10)	Yes	No(2)	Yes	Yes	No(9)	No(4)	No(10)
DBeckham	Yes	No(10)	No(10)	Yes	No(10)	No(4)	Yes	No(10)	No(10)	No(10)
AHathaway	Yes	Yes	Yes	Yes	No(10)	Yes	No(9)	Yes	No(6)	No(2)
AJolie	No(7)	Yes	Yes	No(7)	No(5)	No(2)	No(2)	Yes	No(4)	No(4)
APacino	No(6)	No(10)	No(4)	Yes	Yes	No(2)	Yes	No(6)	Yes	No(1)
ASandler	Yes	No(4)	Yes	Yes	No(9)	Yes	Yes	Yes	Yes	Yes

Table 4.3 Summarized results for 20 pictures training and 10 unknown pictures input each character in manual wacthlist.

	% Recognition success
WSmith	40 %
NPortman	20 %
JDepp	20 %
JAlba	40 %
JAniston	40 %
DBeckham	30 %
AHathaway	60 %
AJolie	30 %
APacino	40 %
ASandler	80 %

Table 4.4 Summarized % recognition results for 20 pictures training and 10 unknown pictures input each character in manual watchlist.

4.2. Automatic watchlist

The aim of this modality function is create a watchlist based on automatic eyes detection. To test the watchlist result it has been considered 2 scenarios to train the system and therefore use this training for recognizing.

4.2.1. 1xPCA Training and recognition (6 characters)

This training is the same from 4.1.1 section. However it is done using automatic watchlist design. On this case we will use the following images:

- Will Smith: 51 pictures (ID = 1).
- Natalie Portman: 74 pictures (ID = 2).
- Johnny Depp: 63 pictures (ID = 3).
- Jessica Alba: 75 pictures (ID = 4).

- Jennifer Aniston: 74 pictures (ID = 5).
- David Beckham: 50 pictures (ID = 6).

The number of pictures are different because the automatic detector has not been able to normalize all them as not all the pictures have been eye right detected (See Annex A to see the entire face and eye detection table for the characters used on this training) . On this example, the wrong eyes detections has been also considered.
The result in this case is wrong. The face is not properly recognized. See figure 4.3

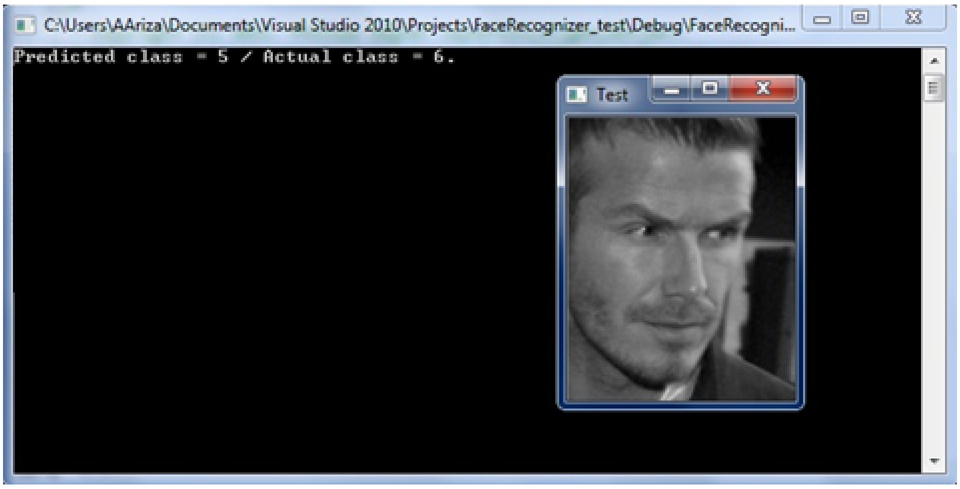

Fig. 4.3 Result for Face Recognition with 1xPCA automatically.

Then lets continue with the same particular test considered on 4.1.1 section with the following images for training:

- Will Smith: 20 pictures (ID = 1)
- Natalie Portman: 20 pictures (ID = 2)
- Johnny Depp: 20 pictures (ID = 3)
- Jessica Alba: 20 pictures (ID = 4)
- Jennifer Aniston: 20 pictures (ID = 5)
- David Beckham: 20 pictures (ID = 6)

And in this case it has been also considered 10 different unknown pictures per character to recognize. The results are summarized on table 4.5.

	Pic 21	Pic 22	Pic 23	Pic 24	Pic 25	Pic 26	Pic 27	Pic 28	Pic 29	Pic 30
WSmith	Yes	No(6)	Yes	No(6)	No(4)	Yes	No(4)	No(2)	No(6)	Yes
NPortman	Yes	No(6)	No(4)	No(6)	No(4)	No(6)	Yes	Yes	No(4)	No(4)
JDepp	No(2)	No(2)	Yes	No(2)	No(4)	No(2)	No(2)	No(1)	No(2)	No(2)
JAlba	No(6)	No(2)	No(1)	Yes	Yes	Yes	No(6)	No(6)	No(6)	No(1)
JAniston	Yes	Yes	No(1)	Yes	No(2)	No(2)	Yes	No(6)	No(4)	Yes
DBeckham	No(1)	No(3)	Yes	No(2)	Yes	Yes	Yes	Yes	Yes	Yes

Table 4.5 Summarized results for 20 pictures training and 10 unknown pictures input each character in automatic watchlist.

	% Recognition success
WSmith	40%
NPortman	30%
JDepp	10%
JAlba	30%
JAniston	50%
DBeckham	70%

Table 4.6 Summarized % recognition results for 20 pictures training and 10 unknown pictures input each character in automatic watchlist.

4.2.2. 1xPCA Training and recognition (10 characters)

This training is the same from 4.1.2 section. However it is done using automatic watchlist design. On this case we will use the following images:

- Will Smith: 51 pictures (ID = 1).
- Natalie Portman: 74 pictures (ID = 2).
- Johnny Depp: 63 pictures (ID = 3).
- Jessica Alba: 75 pictures (ID = 4).
- Jennifer Aniston: 74 pictures (ID = 5).
- David Beckham: 50 pictures (ID = 6).
- Anne Hathaway: 76 pictures (ID = 7).
- Angelina Jolie: 69 pictures (ID = 8).
- Al Pacino: 79 pictures (ID = 9).
- Adam Sandler: 68 pictures (ID = 10).

As it happens on section 4.2.1 the number of pictures are different from 4.1.2 section (in some cases) because the automatic detector has not been able to normalize all them as not all the pictures have been eye right detected (Again, see Annex A to see the entire face and eye detection table for the characters used on this training) .

The processing time of the total operation (train + 1 recognition) is about 5 minutes. The result is not a successful identification of the character. In figure 4.4 it is shown the result where the class (ID) of input image is 10 (Adam Sandler) and the prediction of the recognizer is also ID 6 (David Beckham).

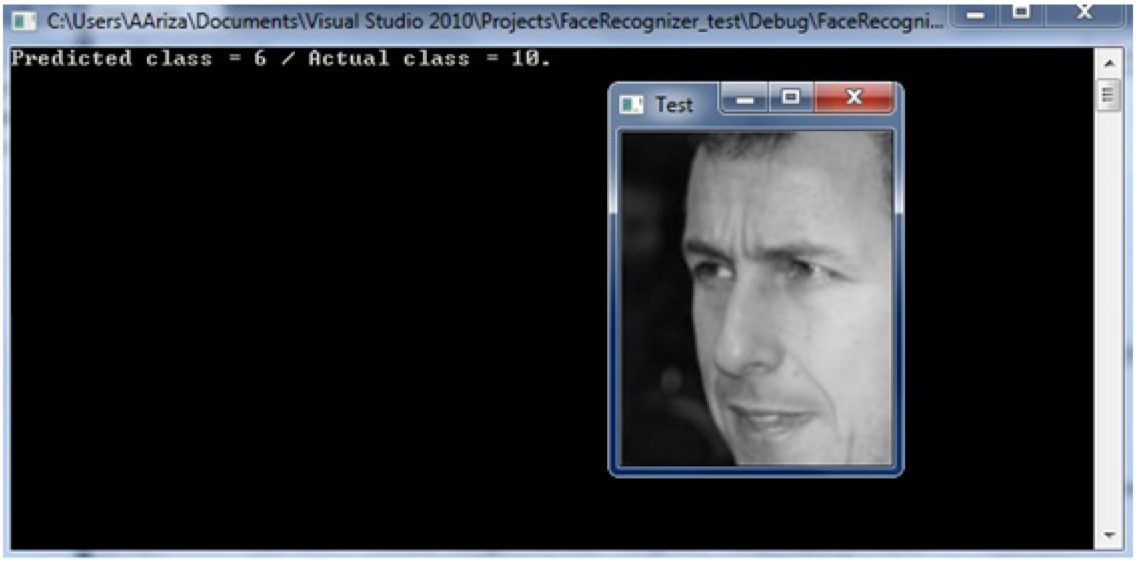

Fig. 4.4 Result for wrong Face Recognition with 1xPCA (10 characters) automatically.

The result of figure 4.4 is given by using the entire watchlist for those 10 characters. So, to minimize times we will train with 20 pictures of each character and we will try to recognize 10 pictures from each one.

Table 4.7 will summarize the results of this training an in 4.8 table the detection percentage will be shown.

	Pic 21	Pic 22	Pic 23	Pic 24	Pic 25	Pic 26	Pic 27	Pic 28	Pic 29	Pic 30
WSmith	No(9)	No(8)	Yes	No(6)	No(4)	No(10)	No(4)	No(2)	No(6)	Yes
NPortman	No(10)	No(8)	No(9)	No(10)	No(6)	No(10)	No(10)	No(2)	No(4)	No(4)
JDepp	No(2)	No(2)	No(9)	No(2)	No(9)	No(2)	Yes	No(1)	No(2)	No(2)
JAlba	No(2)	No(2)	No(9)	Yes	Yes	Yes	No(6)	No(2)	No(8)	No(1)
JAniston	Yes	Yes	No(10)	No(1)	No(2)	No(2)	Yes	No(9)	No(6)	Yes
DBeckham	No(9)	No(10)	Yes	No(10)	No(10)	Yes	No(10)	No(10)	No(10)	No(10)
AHathaway	Yes	Yes	Yes	No(10)	Yes	No(10)	Yes	Yes	Yes	Yes
AJolie	No(7)	No(6)	No(2)	Yes	No(6)	No(10)	No(4)	No(10)	Yes	No(4)
APacino	No(3)	No(4)	No(8)	Yes	No(8)	Yes	Yes	Yes	Yes	Yes
ASandler	Yes	No(1)	No(9)	Yes	Yes	No(7)	Yes	Yes	Yes	Yes

Table 4.7 Summarized results for 20 pictures training and 10 unknown pictures input each character in automatic wacthlist.

	% Recognition success
WSmith	20 %
NPortman	0 %
JDepp	10 %
JAlba	30 %
JAniston	40 %
DBeckham	20 %
AHathaway	80 %
AJolie	20 %
APacino	60 %
ASandler	70 %

Table 4.8 Summarized % recognition results for 20 pictures training and 10 unknown pictures input each character in automatic watchlist.

As it has been seen, the percentage success results are low taking into account the watchlist done in automatically way and using 10 characters.
So, as much as we increase the number of characters included in the watchlist lower is the percentage of recognition success is achieved.

In chapter 5, the conclusions are detailed taking into account the results obtained and the problems faced during the elaboration of the tests.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

Face recognition is a technology that despite the study of it since years ago is nowadays when it has a big importance taking into account the use of it in many applications. And it has been possible because of new hardware appeared (fast processors, high storage capacity, etc.).

The first idea of this thesis is to create a watchlist obtained from an automatic way processing images from a database of images taken from internet. The watchlist will be tested and used to train a system which will recognize persons by entering an unknown image. To see if the recognition is good it is needed to compare the watchlist obtained automatically with a watchlist obtained manually and see if the automatic watchlist works properly or the performance is too low.

With the system properly trained it should be possible to extract information of each camera photogram, detect the face, normalize it and compare it with its database. This application could be useful for the identification and recognition of people searched by police for instance.

Then, this software can be also used for automated systems for vision. Meaning to integrate it in a robot able to recognize people.

Regarding the code itself we have to remark that the automatic detection system doesn't work always with all the image. Because the face and eyes detector don't work with 100% accuracy. The following example shows that it has been detected an eye as an error, so the image has been wrongly rotated and normalized.

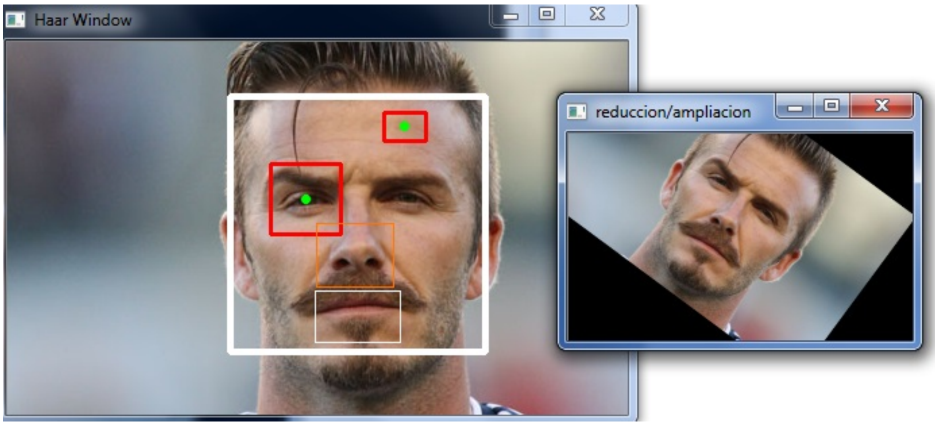

Fig. 5.1 Example of wrong eye detection and wrong normalization of the image.

However another problem is that even the system works fine (detects correctly the eyes) we cannot assure that the image will be correctly normalised setting the eyes centred in X an Y axis because the calculation of the centre of the eyes is no enough accurate. In the next image we can see that the eyes are correctly detected but the centre (in green) of the right bounding box doesn't correspond exactly with the centre of the eye.

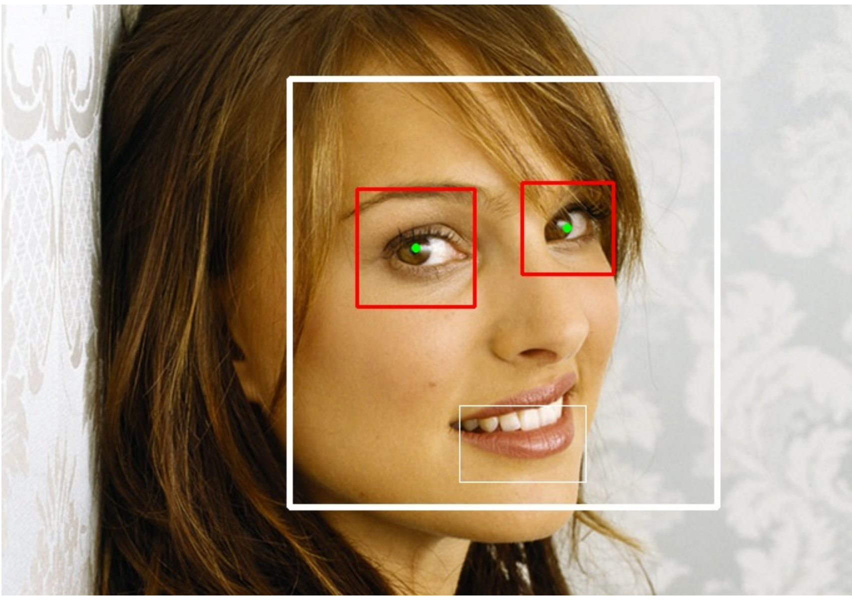

Fig. 5.2 Example of wrong eye centre prediction.

Therefore, despite the manual system to create the database works well we need to be accurate on marking the eye centre, because if not, the system will not put the eyes in the correct X and Y axis for all the pictures.

Regarding the results obtained in chapter 4.It can be compared the results from both databases taking into account the alignment of the eyes

Manual

Fig. 5.3 Example of eyes alignment by manual way database creation.

Automatic

Fig. 5.4 Example of eyes alignment by automatic way database creation.

So, as it is shown it is needed to be as accurate as possible to be able to achieve a good eyes alignment. However for some pictures it has been calculated the error done by marking manually and detecting automatically the centre of the eye (see Annex C to have an idea) and the error is acceptable. Regarding this alignment it is possible to see the accuracy viewing the mean face of all the faces when creating the face space and the projection of each face. In figure 5.5 it is shown the mean face of the pictures from the manual watchlist and in 5.6 for automatic.

Fig. 5.5 Mean face from manual database.

Fig. 5.6 Mean face from automatic database.

As it has been seen in both cases the mean faces have the eyes quite clear (sharp) however the rest of the face is not. This is normal, taking into account that not all the faces have the same view.

Then, on manual watchlist creation, the eyes are clearer that the automatic way. The fact is seen in figures 5.3 and 5.4 where the eyes alignment is better in manual than automatic way.

Regarding the recognition percentage on tests done in 4.1.2 (manual) and 4.2.2 (automatic) sections it has been observed that the percentage is higher in all cases (except in Jennifer Aniston that is the same) for manual than automatically watchlist creation. In the case of Johnny Depp the results are extremely low. However taking a look on those picture it is observed the complexity of them meaning that most of them the face is covered by glasses or long hair.

As a big drawback of the recognizer we can say that it should be trained. This is a task that should be done in a mandatory way because if not the system won't be able to detect anything. This task now is done manually using the .txt files as shown in section 3.6. However it could be improved on a future research unifying the watchlist creation with the training module as well.

Another big drawback of the recognizer implemented right now is the necessity on configuring the proper threshold in order to be the recognizer able to discard images that are not learned. With currently configuration the recognizer will always classify and relate an input image with one of the characters on the database. However the recognizer should be able to classify as unknown if the input image doesn’t correspond with any character.

Regarding future investigation works related with this thesis it is needed to remark that, during the development of this investigation it has been also experimented with nose and mouth detection trying to think in other possibilities of detecting eyes in pictures that are not detected properly. It means on thinking on a detection eyes method based on geometry and taking into account nose and mouth detection to foresee where the eyes should be positioned. This method should be accuracy analysed due to see if the error done by predicting the position of the eyes are low enough. The detection of nose and mouth are shown in tables detections in Annex A for 6 characters.

Therefore it has been thought as future related work on trying to investigate some methods related with achieving an accurate detection of the eye centre. It should be useful due to normalize properly the images.

Finally, the last proposed future work is trying to change the parameters set for the watchlist creation and evaluate its efficiency. The parameters we refer are the size of the images, the distance between eyes, the height of the eyes etc. taking into account that on this thesis those parameters have been established to use an starting logical point.

Personally, I have found this topic very interesting from the beginning. However I needed some time to familiarize me with OpenCV libraries and C++ programming language to achieve the results we have seen. I think the results has been positive, however I hoped having a high successful rate of recognizing. This fact encourages me to continue working on this topic trying to improve the system and trying to obtain a better watchlist able to train a recognizer to obtain high successful recognition rates. And why not trying to use the system to implement an automatic program able to recognize people in photos without the necessity on tagging manually as Facebook does.

BIBLIOGRAPHY

[1] OpenCV software overview. (http://opencv.org/about.html).

[2] Face Recognition definition. (http://en.wikipedia.org/wiki/Face_detection)

[3] Paul Viola and Michael Jones. *"Rapid Object Detection using a Boosted Cascade of Simple"*. Accepted conference on computer vision and pattern recognition, 2001.

[**4**] Pei-Yu Chao, *"Haar Transform and Its Applications"* . Time frequency analysis and wavelet transform tutorial.

[5] Ricoh Technology. *"Weak and Strong Classifiers"*. 2006.

[6] Raúl Rojas. "*AdaBoost and the Super Bowl of Classiers. A Tutorial Introduction to Adaptive Boosting*". Computer Science Department Freie Universit at Berlin. 2009 (http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf).

[7] Yoav Freund and Robert E. Schapire. " *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*". AT&T Labs. 1996

[8] David H. Hubel Biography. (http://www.nobelprize.org/nobel_prizes/medicine/laureates/1981/hubel-bio.html)

[9] Torsten Wiesel Biography. (http://www.faqs.org/health/bios/79/Torsten-Wiesel.html)

[10] Kanade, T. "*Picture processing system by computer complex and recognition of human faces*". PhD thesis, Kyoto University, November 1973

[11] Brunelli, R., Poggio, T. "*Face Recognition through Geometrical Features".* European Conference on Computer Vision (ECCV) 1992, S. 792–800.

[12] Turk, M., and Pentland, A. "*Eigenfaces for recognition".* Journal of Cognitive Neuroscience 3 (1991), 71–86.

[13] Belhumeur, P. N., Hespanha, J., and Kriegman, D. "*Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection".* IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 7 (1997), 711–720.

[14] Timo Ahonen, Abdenour Hadid, and Matti Pietik¨ainen *" Face Recognition with Local Binary Patterns"* University of Oulu, Finland.

ANNEX A

The following tables shows the detailed results of the detections done by the classifiers in the automatically watchlist creation mode. Here, it has been also taken into account the detections of mouth and nose (experimentally done). If the mentioned part is detected the cell contains "YES" and if not "NO" is entered. If more than one part is detected "2x" will appear in the cell, meaning that one or both are right detections. However if it has 1 detection but it is wrong "1xError" is entered into the cell.

Detection of face, eyes, mouth and nose

Num pictures = 56
DataBase = Will Smith

Pictures	Face	Left eye	Right eye	Mouth	Nose
will_smith/will_smith-1.JPG	YES	YES	YES	YES	2x
will_smith/will_smith-2.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-3.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-4.JPG	YES	YES	YES	YES	2x
will_smith/will_smith-5.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-6.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-7.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-8.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-9.JPG	YES	YES	YES	NO	NO
will_smith/will_smith-10.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-17.JPG	YES	YES	YES	YES	NO
will_smith/will_smith-18.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-19.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-20.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-21.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-24.JPG	YES	YES	YES	1xError	YES
will_smith/will_smith-27.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-28.JPG	NO	NO	NO	NO	NO
will_smith/will_smith-29.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-30.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-33.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-34.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-35.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-36.JPG	YES	YES	1xError	YES	NO
will_smith/will_smith-37.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-38.JPG	YES	YES	1xError	YES	NO
will_smith/will_smith-39.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-40.JPG	YES	YES	YES	YES	YES

will_smith/will_smith-41.JPG	YES	YES	YES	YES	NO
will_smith/will_smith-43.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-44.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-46.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-47.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-50.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-52.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-53.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-54.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-55.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-56.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-57.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-59.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-60.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-61.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-62.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-63.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-64.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-68.JPG	YES	YES	YES	NO	YES
will_smith/will_smith-72.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-73.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-74.JPG	YES	YES	YES	YES	2x
will_smith/will_smith-79.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-80.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-83.JPG	YES	YES	YES	YES	NO
will_smith/will_smith-84.JPG	YES	YES	YES	YES	YES
will_smith/will_smith-86.JPG	1xError	NO	NO	NO	NO
will_smith/will_smith-88.JPG	YES	YES	YES	YES	YES

Fig A.1 Will Smith table detection

Results - Will Smith	
% Face detection	**96,4%**
% Left eye detection	**96,4%**
% Right eye detection	**92,9%**
% Mouth detection	**91,1%**
% Nose detection	**80,4%**
% Both eyes detection	**92,9%**

Fig A.2 Will Smith table summary detection

Detection of face, eyes, mouth and nose

Num pictures = 76
DataBase = Jessica Alba

Pictures	Face	Left eye	Right eye	Mouth	Nose
jessica_alba/jessica_alba-1.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-2.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-3.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-4.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-5.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-6.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-7.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-8.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-9.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-10.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-11.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-12.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-13.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-14.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-15.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-16.JPG	2x	YES	YES	YES	YES
jessica_alba/jessica_alba-17.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-18.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-19.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-20.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-21.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-23.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-24.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-25.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-26.JPG	YES	YES	YES	YES	NO
jessica_alba/jessica_alba-27.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-28.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-29.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-30.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-32.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-33.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-34.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-35.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-36.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-37.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-38.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-39.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-41.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-42.JPG	YES	YES	YES	YES	YES

jessica_alba/jessica_alba-43.JPG	2x	YES	YES	YES	YES
jessica_alba/jessica_alba-46.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-47.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-48.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-49.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-52.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-53.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-54.JPG	YES	YES	YES	YES	NO
jessica_alba/jessica_alba-56.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-58.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-59.JPG	YES	YES	YES	YES	1xError
jessica_alba/jessica_alba-60.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-62.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-64.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-65.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-66.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-67.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-68.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-69.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-70.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-71.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-73.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-74.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-75.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-76.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-77.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-78.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-79.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-80.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-81.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-82.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-83.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-85.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-86.JPG	YES	YES	YES	YES	2x
jessica_alba/jessica_alba-87.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-88.JPG	YES	YES	YES	YES	YES
jessica_alba/jessica_alba-90.JPG	YES	YES	YES	YES	YES

Fig A.3 Jessica Alba table detection

Results - Jessica Alba	
% Face detection	97,4%
% Left eye detection	100,0%
% Right eye detection	100,0%
% Mouth detection	100,0%
% Nose detection	85,5%
% Both eyes detection	100,0%

Fig A.4 Jessica Alba table summary detection

Detection of face, eyes, mouth and nose
Num pictures = 52
DataBase = David Beckham

Pictures	Face	Left eye	Right eye	Mouth	Nose
david_beckham/david_beckham-1.JPG	YES	YES	YES	NO	YES
david_beckham/david_beckham-2.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-3.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-4.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-5.JPG	YES	YES	YES	YES	2x
david_beckham/david_beckham-6.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-8.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-9.JPG	YES	YES	YES	NO	YES
david_beckham/david_beckham-10.JPG	YES	YES	YES	YES	2x
david_beckham/david_beckham-11.JPG	YES	YES	YES	NO	NO
david_beckham/david_beckham-12.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-13.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-15.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-16.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-17.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-18.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-20.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-21.JPG	YES	YES	1xError	YES	YES
david_beckham/david_beckham-23.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-30.JPG	YES	YES	NO	NO	NO
david_beckham/david_beckham-31.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-34.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-36.JPG	YES	NO	YES	YES	NO
david_beckham/david_beckham-38.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-39.JPG	YES	YES	YES	YES	2x
david_beckham/david_beckham-44.JPG	YES	YES	YES	NO	YES
david_beckham/david_beckham-47.JPG	YES	YES	YES	YES	YES

david_beckham/david_beckham-49.JPG	YES	YES	YES	YES	NO
david_beckham/david_beckham-52.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-54.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-56.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-57.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-58.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-61.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-62.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-64.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-67.JPG	YES	YES	YES	1xError	YES
david_beckham/david_beckham-68.JPG	YES	YES	YES	YES	NO
david_beckham/david_beckham-69.JPG	YES	YES	YES	YES	SI
david_beckham/david_beckham-71.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-72.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-73.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-74.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-75.JPG	YES	YES	YES	YES	NO
david_beckham/david_beckham-76.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-77.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-79.JPG	YES	YES	YES	YES	2x
david_beckham/david_beckham-80.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-81.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-82.JPG	YES	YES	YES	YES	NO
david_beckham/david_beckham-88.JPG	YES	YES	YES	YES	YES
david_beckham/david_beckham-89.JPG	YES	YES	YES	YES	YES

Fig A.5 David Beckham table detection

Results - David Beckham	
% Face detection	100,0%
% Left eye detection	98,1%
% Right eye detection	96,2%
% Mouth detection	88,5%
% Nose detection	78,8%
% Both eyes detection	96,2%

Fig A.6 David Beckham table summary detection

Detection of face, eyes, mouth and nose

Num pictures = 70
DataBase = Johnny Depp

Pictures	Face	Left eye	Right eye	Mouth	Nose
johnny_depp/johnny_depp-1.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-2.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-3.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-4.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-5.JPG,	YES	YES	YES	YES	2x
johnny_depp/johnny_depp-6.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-7.JPG,	NO	NO	NO	NO	NO
johnny_depp/johnny_depp-9.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-10.JPG,	YES	YES	YES	YES	2x
johnny_depp/johnny_depp-11.JPG,	YES	YES	YES	YES	2x
johnny_depp/johnny_depp-12.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-13.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-14.JPG,	1xError	NO	YES	1xError	YES
johnny_depp/johnny_depp-15.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-16.JPG,	1xError	NO	NO	NO	NO
johnny_depp/johnny_depp-17.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-18.JPG,	YES	YES	YES	YES	2x
johnny_depp/johnny_depp-19.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-21.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-22.JPG,	YES	YES	YES	NO	NO
johnny_depp/johnny_depp-23.JPG,	YES	YES	YES	1xError	YES
johnny_depp/johnny_depp-24.JPG,	YES	YES	NO	YES	YES
johnny_depp/johnny_depp-27.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-28.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-29.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-30.JPG,	YES	NO	YES	YES	2x
johnny_depp/johnny_depp-32.JPG,	YES	YES	NO	YES	YES
johnny_depp/johnny_depp-33.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-35.JPG,	YES	YES	NO	YES	YES
johnny_depp/johnny_depp-36.JPG,	YES	YES	YES	NO	YES
johnny_depp/johnny_depp-38.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-39.JPG,	YES	YES	YES	YES	2x
johnny_depp/johnny_depp-42.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-45.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-46.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-47.JPG,	YES	YES	YES	1xError	YES
johnny_depp/johnny_depp-48.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-49.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-50.JPG,	YES	YES	YES	YES	YES

johnny_depp/johnny_depp-51.JPG,	YES	YES	YES	2x	YES
johnny_depp/johnny_depp-52.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-53.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-54.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-55.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-56.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-57.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-58.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-59.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-61.JPG,	2x	YES	YES	YES	YES
johnny_depp/johnny_depp-62.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-63.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-64.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-66.JPG,	2x	2x	2x	2x	YES
johnny_depp/johnny_depp-67.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-68.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-70.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-71.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-72.JPG,	YES	YES	1xError	YES	YES
johnny_depp/johnny_depp-75.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-76.JPG,	1xError	1xError	YES	YES	YES
johnny_depp/johnny_depp-77.JPG,	YES	YES	YES	NO	YES
johnny_depp/johnny_depp-78.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-79.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-81.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-82.JPG,	1xError	NO	NO	NO	NO
johnny_depp/johnny_depp-86.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-89.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-90.JPG,	YES	YES	YES	YES	YES
johnny_depp/johnny_depp-92.JPG,	1xError	NO	NO	NO	NO
johnny_depp/johnny_depp-93.JPG,	1xError	NO	NO	NO	NO

Fig A.7 Johnny Depp table detection

Results - Johnny Depp	
% Face detection	87,1%
% Left eye detection	87,1%
% Right eye detection	85,7%
% Mouth detection	81,4%
% Nose detection	81,4%
% Both eyes detection	81,4%

Fig A.8 Johnny Depp table summary detection

Detection of face, eyes, mouth and nose

Num pictures = 77
DataBase = Natalie Portman

Pictures	Face	Left eye	Right eye	Mouth	Nose
natalie_portman/natalie_portman-1.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-3.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-4.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-6.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-7.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-8.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-9.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-10.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-11.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-12.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-13.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-14.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-16.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-17.JPG,	YES	NO	YES	YES	YES
natalie_portman/natalie_portman-18.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-19.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-20.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-21.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-22.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-23.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-26.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-27.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-29.JPG,	YES	YES	YES	YES	NO
natalie_portman/natalie_portman-30.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-31.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-32.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-33.JPG,	YES	YES	YES	YES	YES

natalie_portman/natalie_portman-36.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-37.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-38.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-39.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-40.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-41.JPG,	YES	YES	YES	YES	NO
natalie_portman/natalie_portman-42.JPG,	YES	YES	YES	YES	NO
natalie_portman/natalie_portman-43.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-44.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-45.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-46.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-49.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-50.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-51.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-52.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-53.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-54.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-56.JPG,	YES	YES	YES	YES	NO
natalie_portman/natalie_portman-57.JPG,	YES	YES	Lateral	Lateral	Lateral
natalie_portman/natalie_portman-58.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-59.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-60.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-61.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-62.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-63.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-65.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-66.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-68.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-69.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-70.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-71.JPG,	YES	YES	YES	YES	2x
natalie_portman/natalie_portman-72.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-73.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-74.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-75.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-76.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-78.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-79.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-80.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-81.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-82.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-83.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-84.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-85.JPG,	YES	YES	YES	YES	YES

natalie_portman/natalie_portman-87.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-88.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-89.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-90.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-91.JPG,	YES	YES	YES	YES	YES
natalie_portman/natalie_portman-92.JPG,	YES	YES	YES	YES	YES

Fig A.9 Natalie Portman table detection

Results - Natalie Portman	
% Face detection	100,0%
% Left eye detection	98,7%
% Right eye detection	98,7%
% Mouth detection	98,7%
% Nose detection	75,3%
% Both eyes detection	97,4%

Fig A.10 Natalie Portman table summary detection

Detection of face, eyes, mouth and nose

Num pictures = 76

DataBase = Jennifer Aniston

Pictures	Face	Left eye	Right eye	Mouth	Nose
jennifer_aniston/jennifer_aniston-1.JPG,	YES	YES	YES	YES	NO
jennifer_aniston/jennifer_aniston-2.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-3.JPG,	YES	YES	YES	NO	YES
jennifer_aniston/jennifer_aniston-4.JPG,	YES	YES	YES	YES	NO
jennifer_aniston/jennifer_aniston-5.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-6.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-7.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-9.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-10.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-11.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-12.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-13.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-14.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-15.JPG,	YES	YES	YES	1xError	YES
jennifer_aniston/jennifer_aniston-16.JPG,	YES	YES	YES	NO	NO
jennifer_aniston/jennifer_aniston-18.JPG,	YES	YES	YES	NO	YES

jennifer_aniston/jennifer_aniston-19.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-20.JPG,	YES	YES	YES	1xError	YES
jennifer_aniston/jennifer_aniston-21.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-22.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-23.JPG,	YES	YES	YES	NO	NO
jennifer_aniston/jennifer_aniston-25.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-26.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-27.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-28.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-29.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-30.JPG,	YES	YES	YES	NO	YES
jennifer_aniston/jennifer_aniston-31.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-32.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-33.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-35.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-36.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-37.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-38.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-39.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-41.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-42.JPG,	YES	YES	YES	2x	YES
jennifer_aniston/jennifer_aniston-44.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-45.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-46.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-47.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-48.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-49.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-50.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-51.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-53.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-54.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-55.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-56.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-57.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-58.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-59.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-60.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-61.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-63.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-64.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-65.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-66.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-67.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-68.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-69.JPG,	YES	YES	YES	YES	2x

jennifer_aniston/jennifer_aniston-70.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-71.JPG,	YES	NO	YES	YES	YES
jennifer_aniston/jennifer_aniston-72.JPG,	YES	YES	YES	YES	2x
jennifer_aniston/jennifer_aniston-73.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-74.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-75.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-76.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-77.JPG,	YES	YES	YES	1xError	YES
jennifer_aniston/jennifer_aniston-78.JPG,	1xError	1xError	1xError	1xError	1xError
jennifer_aniston/jennifer_aniston-79.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-80.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-81.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-83.JPG,	YES	YES	YES	YES	YES
jennifer_aniston/jennifer_aniston-86.JPG,	1xError	NO	NO	NO	NO
jennifer_aniston/jennifer_aniston-87.JPG,	YES	NO	YES	NO	NO

Fig A.11 Natalie Portman table detection

Results - Jennifer Aniston	
% Face detection	**97,4%**
% Left eye detection	**93,4%**
% Right eye detection	**97,4%**
% Mouth detection	**84,2%**
% Nose detection	**81,6%**
% Both eyes detection	**92,1%**

Fig A.12 Natalie Portman table summary detection

ANNEX B

In this Annex, it is compared the use of 1 classifier (haarcascade_eye.xml) for eyes and the use of 4 filters in cascade for 3 characters: Will Smith, Jessica Alba and David Beckham.
In the face detector case, the comparison is done between the use of 1 (haarcascade_frontalface_alt.xml) classifier and 3 for the mentioned 3 characters.

Face Detection

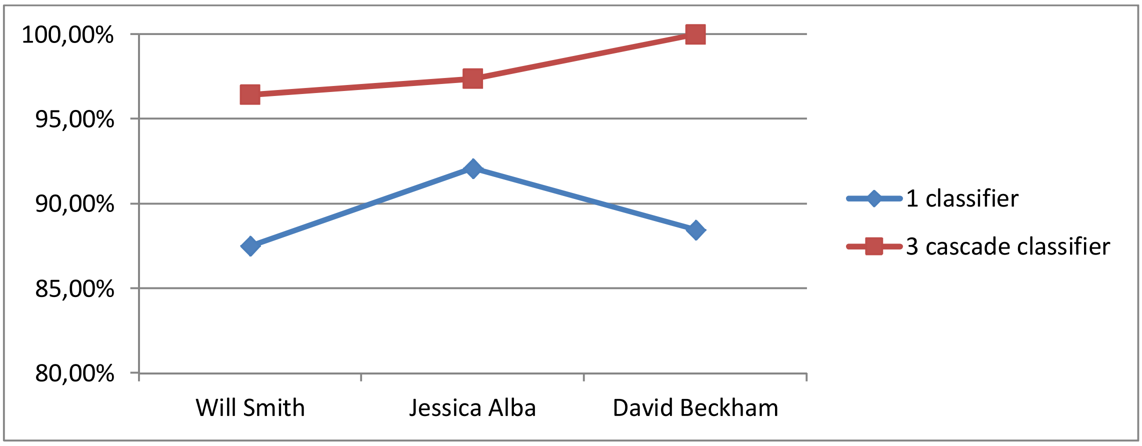

Fig B.1 3 Character detection percentage using 1 or 3 cascade classifiers for face detection.

Left eye detection

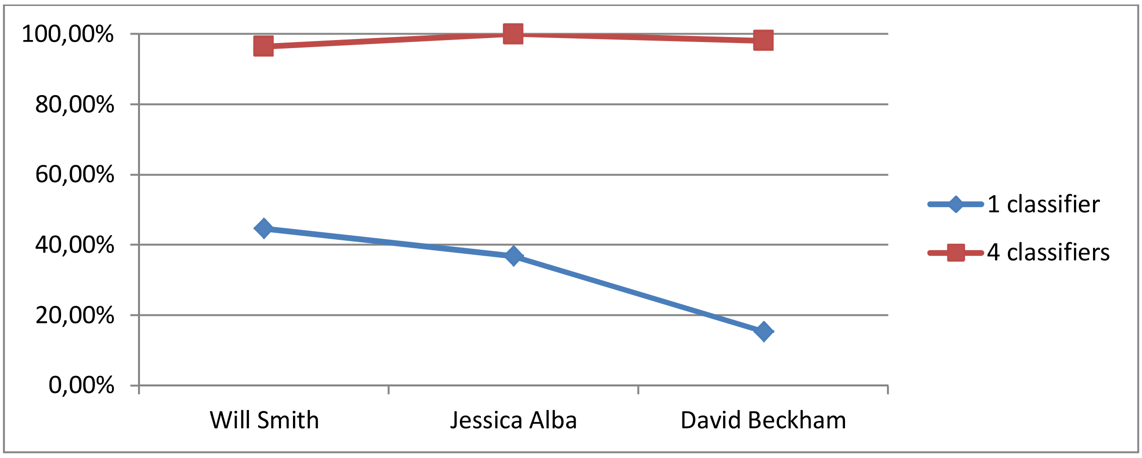

Fig B.2 3 Character detection percentage using 1 or 4 cascade classifiers for left eye detection.

Right eye detection

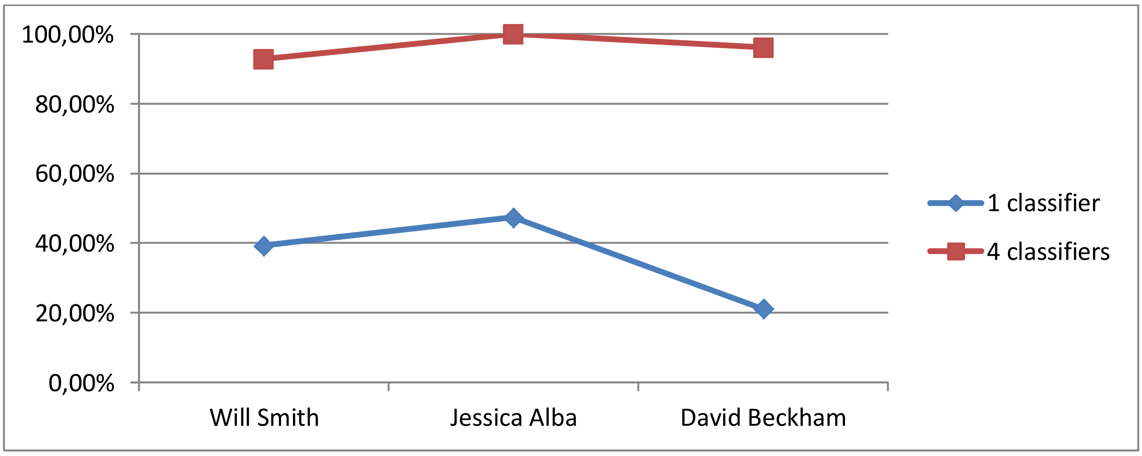

Fig B.3 3 Character detection percentage using 1 or 4 cascade classifiers for right eye detection.

Both eyes detection

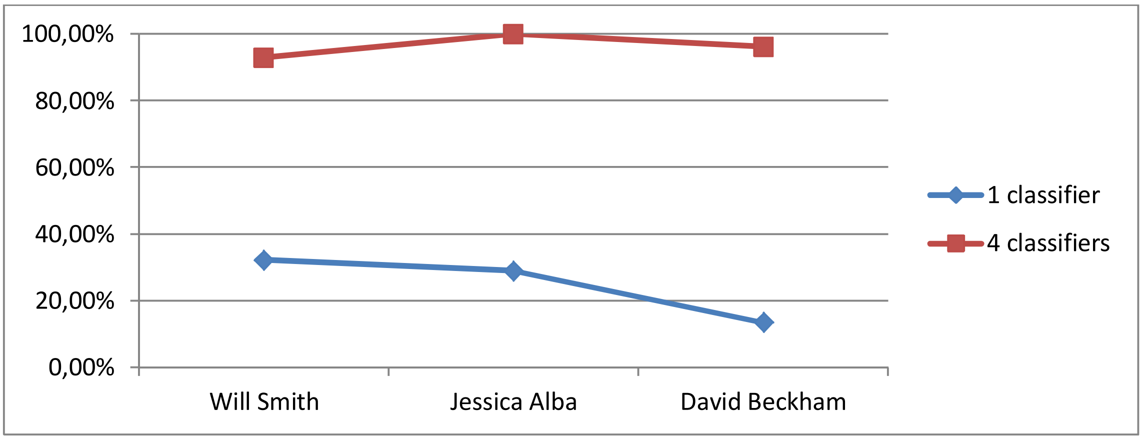

Fig B.4 3 Character detection percentage using 1 or 4 cascade classifiers for eyes detection.

ANNEX C

The following error tables are calculated not using the 4 cascade levels for eyes detection but only 1 of them: `haarcascade_eye.xml`

The second and third columns shows the difference error done in absolute value on X and Y axis compared with manual marking and automatic eye detection way. Then the % for the pixel variation is calculated for both and separately eyes. Where the errors are calculated as:

$$\%\ Error\ Left\ Eye = \left(\left(\left(\frac{X\_left_{Coord_{deviation}} pixel}{Bounding\ box\ left\ size}\right) + \left(\frac{Y\_left_{Coord_{deviation}} pixel}{Bounding\ box\ left\ size}\right)\right)/2\right) * 100$$

Formula C.1 % Error calculated for Left eye marking between automatic and manually watchlist creation way.

$$\%\ Error\ Right\ Eye = \left(\left(\left(\frac{X\_right_{Coord_{deviation}} pixel}{Bounding\ box\ right\ size}\right) + \left(\frac{Y\_right_{Coord_{deviation}} pixel}{Bounding\ box\ right\ size}\right)\right)/2\right) * 100$$

Formula C.2 % Error calculated for Right eye marking between automatic and manually watchlist creation way.

$$\%\ Mean\ error = \frac{\%\ Error\ Left\ Eye + \%\ Error\ Right\ Eye}{2}$$

Formula C.3 % Mean Error calculated for both eyes marking between automatic and manually watchlist creation way.

Error in pixels for pictures where 2 eyes are detected							
Number evaluated pictures = 15							
DataBase = Will Smith							
Pictures	Error Left Eye	Error Left Eye	Bounding box Left Eye	Bounding box Right Eye	% Error Left Eye	% Error Right Eye	% Mean Error
Pic 1	(3 , 4)	(1 , 1)	49	52	7,14%	1,92%	4,53%
Pic 2	(3 , 1)	(0 , 0)	49	47	4,08%	0,00%	2,04%
Pic 3	(2 , 1)	(1 , 0)	46	49	3,26%	1,02%	2,14%
Pic 4	(2 , 3)	(2 , 2)	61	62	4,10%	3,23%	3,66%
Pic 5	(2 , 1)	(2 , 1)	48	51	3,13%	2,94%	3,03%
Pic 6	(3 , 2)	(4 , 1)	71	59	3,52%	4,24%	3,88%
Pic 8	(3 , 2)	(1 , 0)	54	49	4,63%	1,02%	2,83%
Pic 9	(0 , 2)	(1 , 0)	47	48	2,13%	1,04%	1,58%
Pic 10	(2 , 1)	(1 , 1)	54	56	2,78%	1,79%	2,28%
Pic 12	(3 , 2)	(0 , 1)	46	45	5,43%	1,11%	3,27%
Pic 13	(2 , 1)	(3 , 1)	44	45	3,41%	4,44%	3,93%
Pic 14	(3 , 1)	(2, 0)	54	45	3,70%	2,22%	2,96%
Pic 15	(2 , 3)	(0 , 2)	64	68	3,91%	1,47%	2,69%

Fig C.1 Will Smith calculation eyes error comparing images automatically detected and manually marked.

Error in pixels for pictures where 2 eyes are detected

Number evaluated pictures = 19
DataBase = Jessica Alba

Pictures	Error Left Eye	Error Left Eye	Bounding box Left Eye	Bounding box Right Eye	% Error Left Eye	% Error Right Eye	% Mean Error
Pic 1	(3 , 2)	(0, 2)	52	46	4,81%	2,17%	3,49%
Pic 2	(2 , 4)	(0, 1)	51	48	5,88%	1,04%	3,46%
Pic 3	(6 , 0)	(2, 2)	50	56	6,00%	3,57%	4,79%
Pic 4	(1 , 2)	(1, 2)	55	54	2,73%	2,78%	2,75%
Pic 5	(2 , 0)	(1, 2)	48	48	2,08%	3,13%	2,60%
Pic 6	(5 , 1)	(0 , 0)	45	51	6,67%	0,00%	3,33%
Pic 7	(4 , 0)	(5 , 3)	67	62	2,99%	6,45%	4,72%
Pic 8	(2 , 1)	(0 , 2)	55	54	3,64%	1,85%	2,74%
Pic 9	(1 , 1)	(1 , 1)	43	46	2,33%	2,17%	2,25%
Pic 10	(4 , 1)	(0 , 1)	45	51	5,56%	0,98%	3,27%
Pic 11	(0 , 1)	(9 , 2)	52	55	0,96%	10,00%	5,48%
Pic 12	(2 , 1)	(2 , 0)	53	51	2,83%	1,96%	2,40%
Pic 13	(1 , 3)	(3 , 1)	45	50	4,44%	4,00%	4,22%
Pic 14	(2 , 1)	(0 , 2)	51	52	2,94%	1,92%	2,43%

Fig C.2 Jessica Alba calculation eyes error comparing images automatically detected and manually marked.

Error in pixels for pictures where 2 eyes are detected

Number evaluated pictures = 7
DataBase = David Beckham

Pictures	Error Left Eye	Error Left Eye	Bounding box Left Eye	Bounding box Right Eye	% Error Left Eye	% Error Right Eye	% Mean Error
Pic 1	(3 , 2)	(2, 2)	45	47	5,56%	4,26%	4,91%
Pic 2	(4 , 2)	(1, 2)	53	56	5,66%	2,68%	4,17%
Pic 3	(1 , 2)	(3, 1)	58	53	2,59%	3,77%	3,18%
Pic 4	(1 , 2)	(2, 2)	48	48	3,13%	4,17%	3,65%
Pic 5	(7 , 4)	(12 , 3)	65	91	8,46%	8,24%	8,35%

Fig C.3 David Beckham calculation eyes error comparing images automatically detected and manually marked.