Person Identification for Visually-Controlled Picture-in-Picture Activation with Sound Zones

Daniel Michelsanti

Vision, Graphics and Interactive Systems, 9^{th} semester
Project group 947
Aalborg University

AAU Project Supervisor:
Zheng-Hua Tan

Company Supervisor:
Sven Ewan Shepstone

Thursday, January 12^{th} 2017

AALBORG UNIVERSITET

BANG & OLUFSEN

Daniel Michelsanti

Signature

Date

Vision, Graphics and Interactive Systems 9
Department of Electronic Systems
School of Information and Communication Technology
http://sict.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:

Person Identification for Visually-Controlled Picture-in-Picture Activation with Sound Zones

Theme:

Interactive Systems

Project period and hand-in:

Thursday, September 1^{st} 2016 - Thursday, January 12^{th} 2017

Project group:

947

Participant:

Daniel Michelsanti

AAU Project Supervisor:

Zheng-Hua Tan

Company Supervisor:

Sven Ewan Shepstone

Number of pages: 61

Number of appendices: 4

Synopsis:

A person identification system is a crucial component in many applications from access control to forensics. This work presents a real-time system to recognise an individual in order to offer a personalised experience for television frameworks. Two modalities have been used for this purpose: face, and voice. A comparison between four different techniques have been performed for face recognition on the AT&T database: Eigenfaces, Fisherfaces, Local Binary Patterns Histograms, and Convolutional Neural Networks. A multimodal system consisting of a Convolutional Neural Network for face recognition and i-vectors for speaker recognition has been evaluated on a custom database made of images and speech segments from a number of TED talks.

PREFACE

This report has been made by Daniel Michelsanti, a 9th semester student of the master programme in Vision, Graphics and Interactive Systems at Aalborg University.

The report documents the project-oriented work experience of the student at Bang & Olufsen during the autumn semester of 2016.

The content of this report is confidential due to the information regarding the company and the personal details of the student.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A system for person identification allows to recognise an individual using human characteristics, known as biometrics. They can be divided into two groups: physiological (such as DNA, fingerprint, and face), and behavioural (such as gait, and voice). The use of biometrics is generally preferred to the traditional identification systems, like driver's license or personal identification number, because they are more reliable, even though privacy issues may arise.

The goal of this report is to document the internship experience of the student Daniel Michelsanti at Bang & Olufsen, where he implemented an identification system for an interactive prototype. The idea was to use off-the-shelf libraries to recognise a person through face and voice with a camera and a microphone.

The report consists of two parts:

- Academic part. Here a description of the problem is detailed, along with the theories, the used methods, and the conducted tests.
- Analytical part. It presents an overview of the company and the considerations of the student regarding the internship.

CHAPTER 2

ACADEMIC PART

This part of the report contains a description of the academic issues which the student has focused on during the project-oriented work at Bang & Olufsen.

2.1 Problem Analysis

Nowadays different sensor technologies are available, and they allow to recognise a person and his/her intentions. These technologies can be used to interact with several kinds of devices in a different way, without using a remote control (Figure 2.1). The sensor-based interaction modalities currently on the market do not feel natural yet, in some cases because the technology is not mature and reliable enough, and in others because it is not being used in a way that feels natural. For this reason, Bang & Olufsen would like to explore the best way and the potential for users to interact through voice, gaze, and gestures.

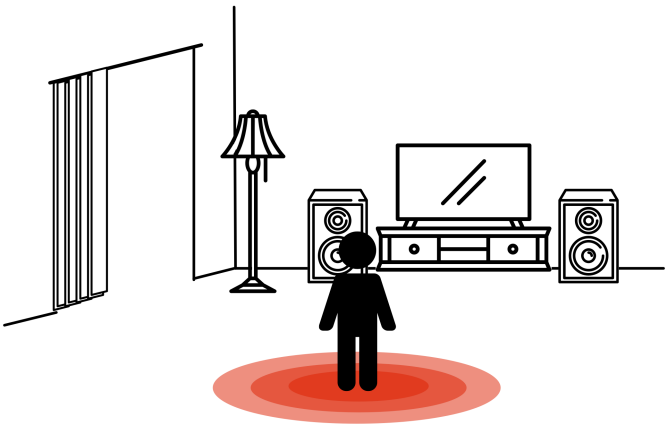


Figure 2.1 Typical scenario where a user needs to interact with multiple systems: the television, the sound system, the lighting, the curtains etc. In this situation the use of sensor technologies can dramatically change the way the user interacts with the different devices.

In order to accomplish this, interactive prototypes to experiment with different scenarios are needed. In this regard, three key research questions have been identified:

1. Is someone trying to interact with the device?
2. Who is interacting with the device?
3. What is the person trying to communicate?

The focus should be on:

- The use case (the three research questions).
- Identifying the right software components.
- Designing with maximum re-use in mind.
- Embedded development.
- Robustness of components to a number of light and noise conditions.
- Writing tests to evaluate the functionality of components.

The problem that this report addresses is the one of person identification for a gaze as hot-word prototype. The system setup (Figure 2.2) is quite simple, and it consists of a device (for example a television) that can give feedbacks based on the data acquired by a camera and a microphone.

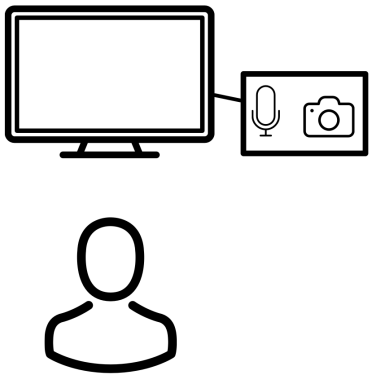


Figure 2.2 System setup. It consists of a device connected to a camera and a microphone.

The use case of the prototype can be described as follows:

1. A person looks into the camera/device.
2. The system detects that the person is looking at it.
3. The system recognises the person.
4. The system changes to a listening mode, giving a feedback to the user in the form of sound and/or changes in the GUI.

5. The user can speak a voice command (e.g. "Turn on the lights" or "Play my favourite playlist") without the need of a keyword (e.g. "OK Google").

6. The system responds by carrying out the action, based on the identity of the user.

2.1.1 Visually-Controlled Picture-in-Picture Activation with Sound Zones

We can apply this concept to the particular case of a TV system, for example to visually control the activation of Picture-in-Picture (PiP) with sound zones.

Picture-in-Picture is a feature that can be found in some televisions, and allows to display two channels at the same time, visually one over the other as shown in Figure 2.3. Currently the main PiP systems available on the market are two, both accessible using the remote control of the television:

1. In the first one [31], the user can set the options of the PiP modality from a window. In particular he/she can select: the channel to display; the size, as one-fourth or one-eighth of the screen, or can select two screens of the same size using half of the screen for each picture; the position of the PiP pop-up, which can be placed in one of the forth corners of the screen; the sound source, between the main screen and the sub-screen. The disadvantage of this system is that, in order to change the PiP options, the user has to access the menu every time.

2. In the second system [34], the PiP mode can be activated by clicking on an icon. The PiP pop-up can be placed wherever on the screen, using the remote. It is also possible to change channel without open a menu, by placing the cursor over the sub-screen and pressing the up/down buttons of the remote.

Figure 2.3 Example of Picture-in-Picture.

The use case of our prototype can be detailed in this way. Two users (A and B) sit in front of the TV. The currently airing channel is Eurosport in the standard mode, that from now on we are going to call *SINGLE PICTURE MODE* (Figure 2.4). In this modality the system continuously detects face and gaze of the users.

Figure 2.4 Single picture mode.

When one of the users (suppose A) performs a particular action (Action 1), such as nodding the head or looking at a corner of the screen for some seconds, then the *GAZE DETECTION MODE* (Figure 2.5) is activated. The system recognises the user, and four hotspots that correspond to the user favourite channels appear at the corners of the screen. We need to highlight that the four displayed channels are based on the identity of the user, so if a different person activates the *GAZE DETECTION MODE*, different channels are shown.

Figure 2.5 Gaze detection mode. When the user activates this mode with an action, four hotspots appear on the screen.

If the system detects a specific action of the user to select one of the hotspots (Action 2), like looking at one of the corners of the TV, then a PiP pop-up of the chosen channel appears: *PIP MODE* is now active (Figure 2.6). The PiP should be

located close to the physical location of the user, for example if he/she is sitting to the left of the screen, then the PiP pop-up will appear somewhere on the left side of the screen. At this point the sound from the PiP pop-up should only reach user A and the background sound should only reach user B.

Figure 2.6 PiP mode.

At this point the system waits for an action from the user, that can be again Action 1 to activate the *GAZE DETECTION MODE*, or a pinch gesture (Action 3), to go to the *SOUND SELECTION MODE* (Figure 2.7). In this modality, the user performs a drop gesture to tell the system to choose the PiP pop-up as main audio/video source in the *SINGLE PICTURE MODE* or a swipe gesture to get rid of the PiP pop-up (Action 4).

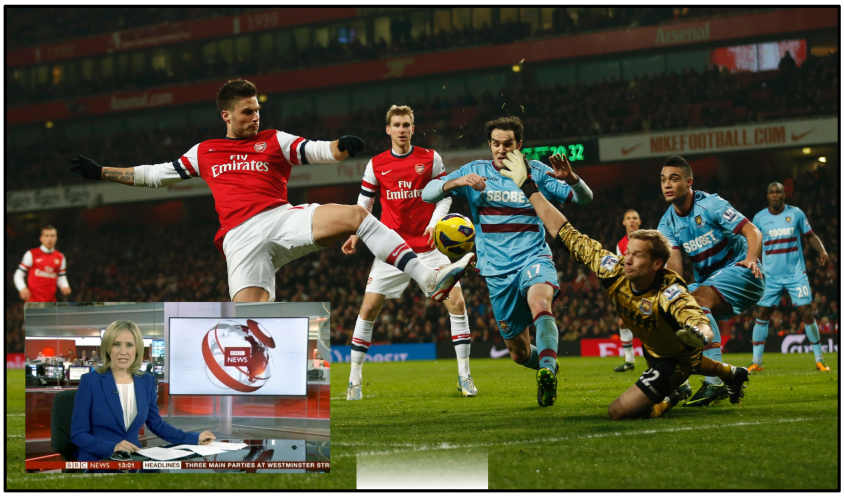


Figure 2.7 Sound selection mode. In this mode the user can drag and drop the PiP pop-up to the hotspot that appears on the bottom of the screen to have it as the main audio/video source. The user can also use a swipe gesture to get rid of the PiP pop-up.

The use case is summarised in the flow chart of Figure 2.8.

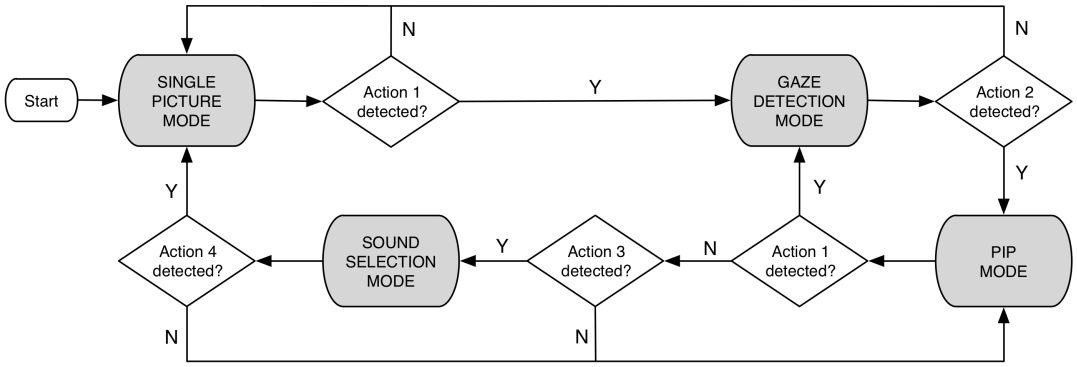


Figure 2.8 Flow chart of the use case for the visually-controlled PiP activation with sound zones.

This prototype has been implemented, but then, in order to highlight the potentialities of a gaze-based system, some modifications have been made. In particular, three corners are used (Figure 2.9): the top-left one to show the information of the current channel; the top-right-one to control the home-automation notifications; the bottom left to show the favourite channel of the user and activate the picture-in-picture. The idea is to trigger specific functionalities when the user looks at one of the corner for a certain amount of time (approximately 3 seconds).

Figure 2.9 The three functionalities that can be activated by gaze in the final prototype.

In order to experiment how this prototype can work we are going to implement the following system (Figure 2.10). A camera captures the video of two users that are watching a television, and for each frame the faces are detected. The crops of the two faces are then used to identify the users, and decide where they are looking based on the gaze detection and the head pose estimation steps, which are not part of this work.

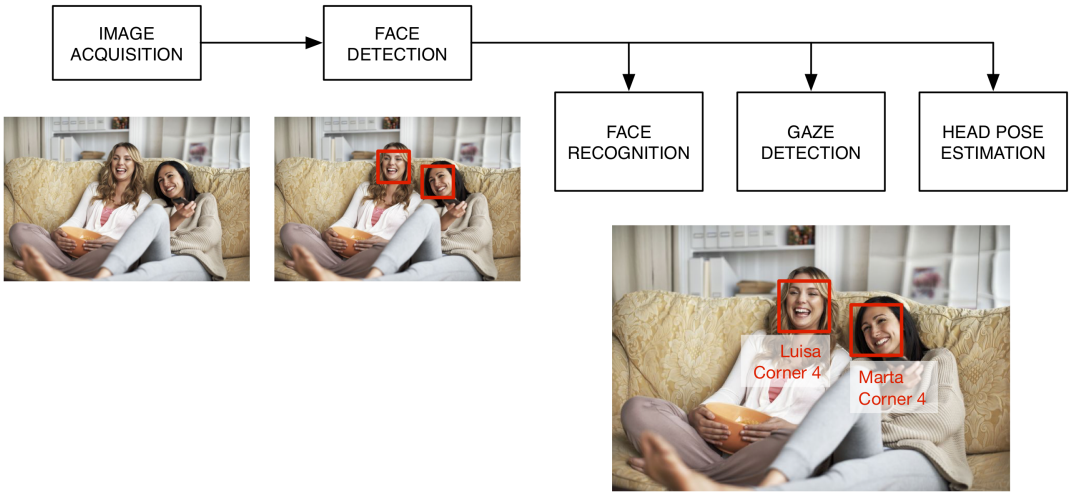


Figure 2.10 Pipeline of the system. For each frame of the camera stream, the image is acquired, then the faces are detected, and they are used to identify the users and decide where they are looking based on the gaze detection and the head pose estimation steps.

2.2 Image Acquisition

For our experiment we use two ways to get a video stream. The first one is by grabbing the stream directly from the built-in camera of our PC. The problem of using this camera is the low resolution of 640×480 pixels. When a person is far away from the camera (more than 2 meters), the details of the face are not really definite, and this makes the gaze hard to be detected. For this reason, the use of a high-resolution camera should be preferred (Figure 2.11). In particular we decided to adopt the camera of a smartphone, that allows to capture a video in higher resolution, in our case 1920×1080 pixels. In order to use the video, we used the application IP Webcam developed by Pavel Khlebovich [23]. We can access the video stream inside a local Wi-Fi network through the following URL: *http://phone_ip_address:port_name/videofeed*.

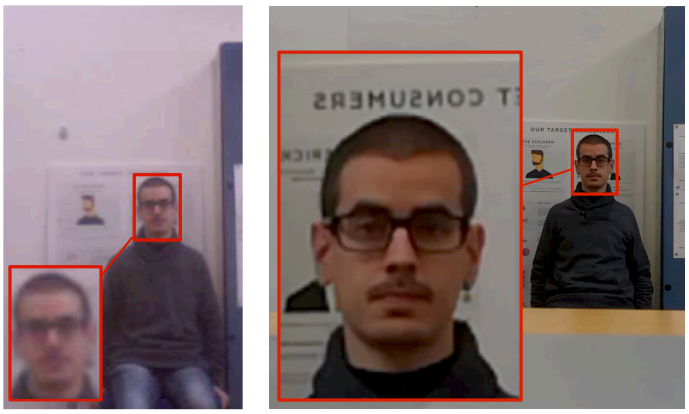


Figure 2.11 Comparison between the data acquired by the two cameras. In this case the person is 3 meters far from the camera. The detail shows the face crop of the 2×-zoomed images.

In the final demonstration an HD Webcam Logitech C615 has been used.

2.3 Face Detection

The problem of face detection can be seen as a specific case of the object detection problem. A possible formulation of this problem is: *given an image, determine the locations and sizes of all the objects belonging to a specific class*. Face detection is so important because of its applications. It is used as first step of automatic face recognition systems, and in some HCI systems that need expression and emotional state recognition. It is still considered a challenging problem because of noise in the image, different poses of the face, problems of occlusion etc. Figure 2.12 shows different conditions in which a face can be captured.

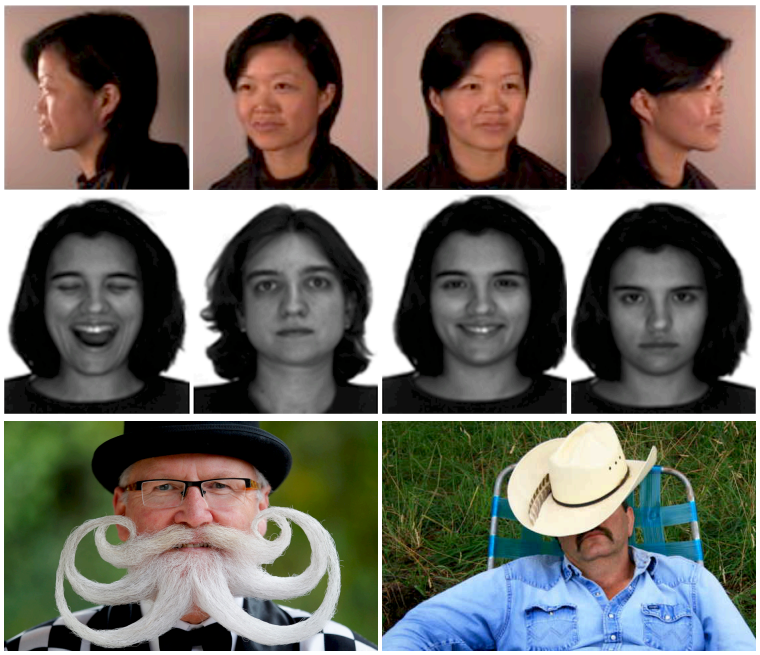


Figure 2.12 Challenges in face detection. A face can be captured in different poses, with different expressions, can have a peculiar shape due to beard, and can be partially occluded.

In order to solve the problem, the paradigm adopted by the vast majority of the researches is the sliding window classification (Figure 2.13). The idea behind this model is simple: the presence of the face is determined by a classifier that uses features extracted from a set of crops of the image. Since the face can be of different sizes, this process is repeated multiple times choosing windows (also known as bounding boxes) of various sizes or several scales of the original image. At the end, redundant boxes around the faces are removed with a non-maximum suppression step.

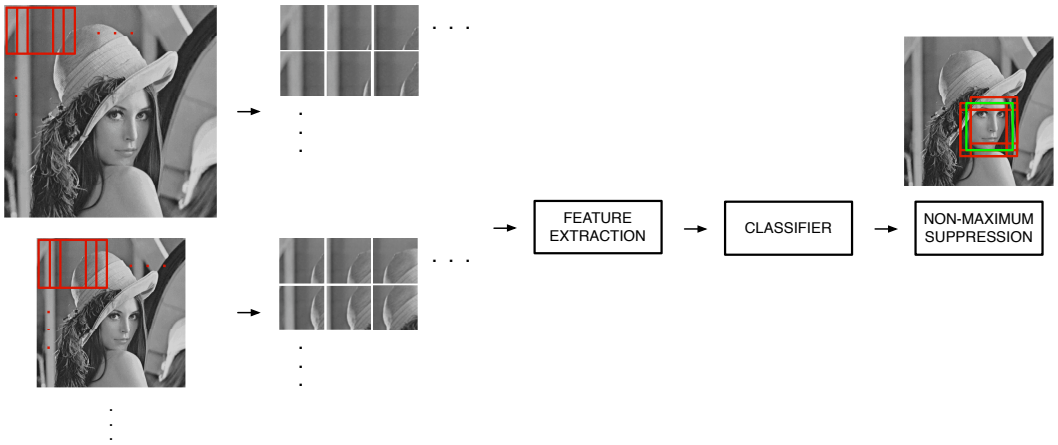


Figure 2.13 Sliding window classification for face detection.

2.3.1 Color-Based Face Detector

One of the simplest method for face detection is based on the skin color (Figure 2.14). The first step is the conversion of the input image into a color space where color information is not mixed with luminance, like HSI (Hue, Saturation, Intensity). This is needed in order to give the system a certain degree of robustness to illumination changes. After this, a probability image is computed, based on a skin model. In particular, the mean and the covariance matrix of many skin patches are computed and used as parameters of a Gaussian distribution. At this point, it is possible to determine the probability that each pixel of the image is a pixel of the skin. Then, a threshold value is used to segment the image, obtaining a binary image in which the pixels set to 1 are considered skin. The next stage consists of a feature extraction on each BLOB (Binary Large OBject). For example, the number of holes, the height-to-width ratio, and the cross-correlation with a face template can be used as input of a classifier in order to determine whether the BLOB is a face.

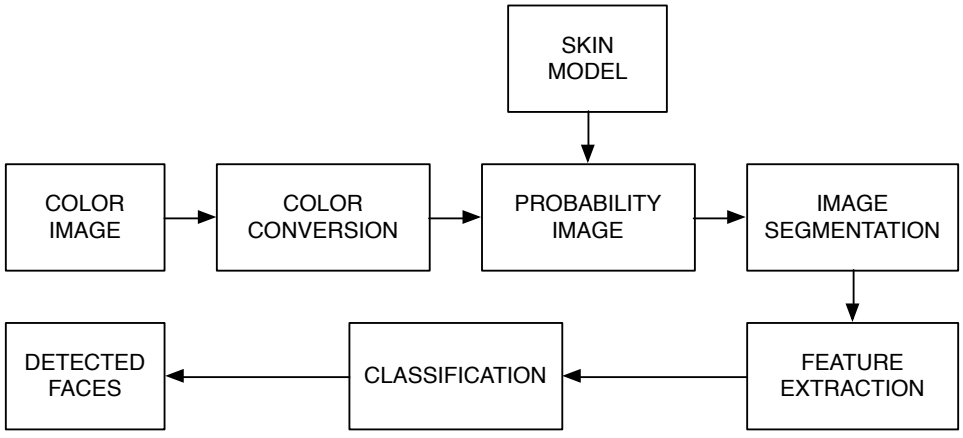


Figure 2.14 Color-based face detection pipeline.

Even though this approach is easy to implement and robust to pose variations, the number of false positives makes it unreliable in several cases.

2.3.2 Viola and Jones Face Detector

A real-time method for face detection that gives better results is the Viola and Jones face detector [35]. It uses the sliding window classification paradigm, and it is based on three concepts:

- *Integral image.* In order to train a classifier that can determine whether a patch of the image is a face, a number of positive and negative samples are needed. From these samples, it is necessary to extract features that can be used by the classifier. In this case rectangular Haar-like features are used. They are obtained by summing up the pixel intensities of different regions of a window and subtracting them. Figure 2.15 shows four examples, where each feature is a value computed by subtracting the sum of the pixels in the white area from the sum of the pixels in the black area.

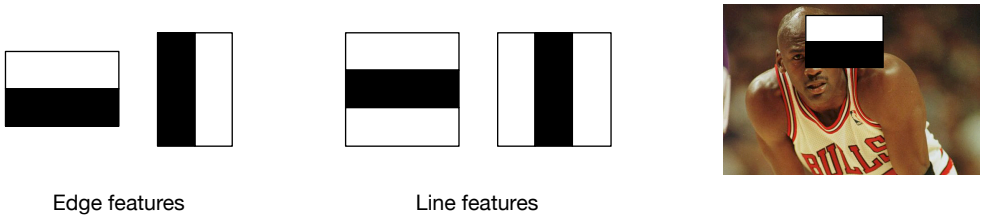


Figure 2.15 Examples of Haar-like features.

The problem is that summing up the intensities of the pixels for each region is computationally expensive if we consider all the possible sizes and locations of each rectangle. For this reason, the calculations are simplified by using the integral image, where a generic pixel at the position (x, y) has the following value:

$$I(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} i(x', y')$$

Using the integral image, calculating the sum within a rectangle consists of an operation that involves only four pixels (the corners of the rectangle).

- *Boosting.* Even if we have a fast method to compute features, the number of possible rectangle features for the detection region is still large, and an evaluation of the whole feature set at test time is inefficient. For this reason, a way to select the most important features is by using Adaboost. It is a classification scheme consisting of a combination of weak learners. At the beginning of the training, each image has an equal weight. Each rectangle feature is evaluated on each training image, and the best threshold that allows to classify faces is selected. Because of errors due to misclassification, features with minimum error rate are selected and the samples reweighted. The process goes on until a certain accuracy rate is reached. The final classifier is a linear combination of these weak learners. The two most important features selected by boosting are the ones in Figure 2.16.

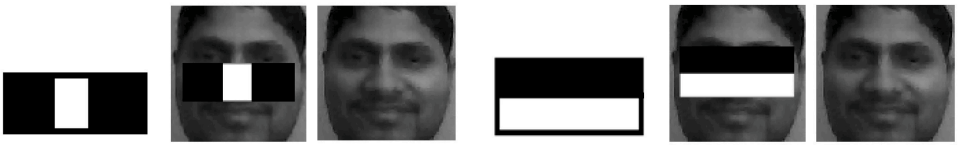


Figure 2.16 Best features selected by boosting. The first one looks like the bridge of the nose, while the second one is similar to the eye region which is darker than the cheeks.

- *Attentional cascade.* At this point, we should classify each window using the most important features found by Adaboost. If we consider the first 200 features, for example, then this solution will be inefficient. In order to reduce the complexity, we can use a simple classifier that rejects the most obvious negative samples, and then use the windows classified as faces as input of a more complex classifier, and so on (Figure 2.17). If a window passes all the steps, then it is recognised as a face.

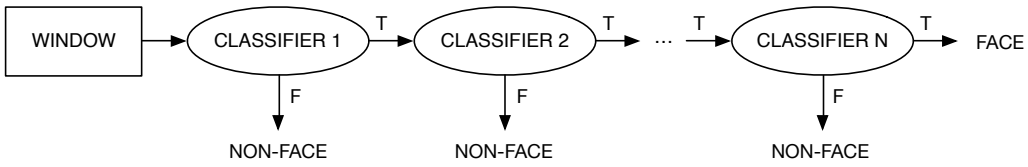


Figure 2.17 N-stage cascade of classifiers.

In their final setup, Viola and Jones used a 38-layer cascade of classifiers which used 6060 features. The training set consisted of 4916 faces, rescaled to 24×24 pixels and around 350 million non-face samples. The training was on the order of weeks on a 466 MHz Sun workstation. With this system a 384×288 pixel image can be processed in around 0.067 s.

2.3.3 HOG and SVM Face Detector

Another possibility for face detection is by using Histogram of Oriented Gradients (HOG) features combined with a linear classifier, such as a Support Vector Machine (SVM). HOG features have been introduced by Dalal and Triggs [13] for human detection, but it can be applied also in other domains. The basic idea is that edges define an object, so if we quantify them in a clever way, we can classify based on it. The pipeline of the detector can be seen in Figure 2.18 and consists of:

- Color normalization. It is applied in order to have a system robust to illumination changes.
- Gradient computation. The gradient of the image is computed by using simple 1-D $[-1, 0, 1]$ derivative masks.
- Weighted voting into spatial and orientation cells. From the horizontal and vertical gradients, it is possible to compute gradient orientations in cells, in order to have a certain number of histograms for each window.

- Contrast normalization over overlapping spatial blocks. The normalization is done in blocks that span multiple cells, for example by using L2-normalization.
- HOGs collection over detection window. All the normalized vector are then concatenated.
- Linear SVM classification. We can use a linear classifier to determine whether the object is the one we want to detect. Generally the used classifier is a SVM, a discriminant classifier that chooses the optimal hyperplane, which is the one that maximizes the margin between the classes.

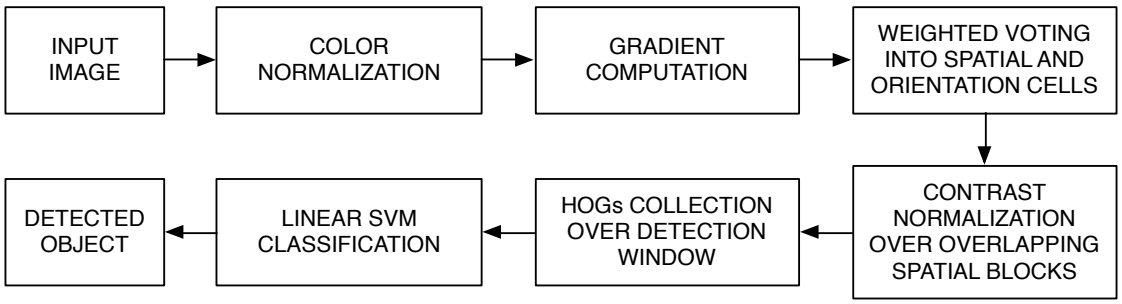


Figure 2.18 Pipeline of a HOG detector as presented in [13].

2.3.4 CNN-Based Face Detector

The current state of the art for face detection is represented by deep convolutional neural networks (CNN). In their work, Farfade et al. [18] fine-tuned AlexNet [24], whose architecture is shown in Figure 2.19, with 200 thousand positive samples and 20 million negative samples, and obtained a detector able to handle different angles of the face and occlusion to some extent.

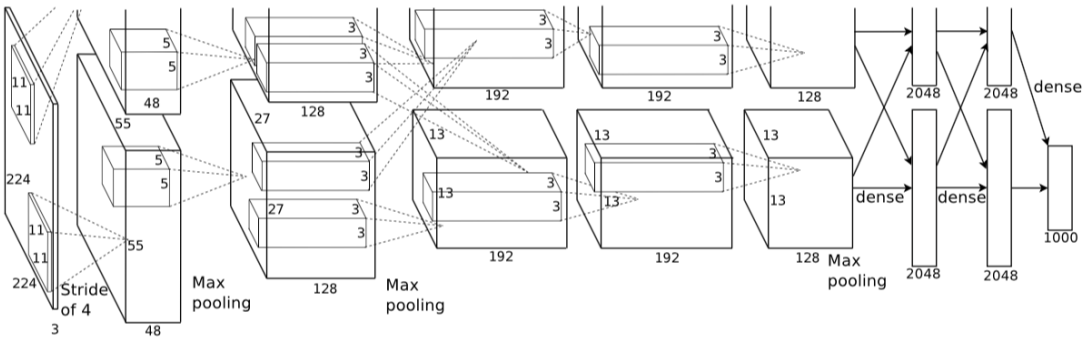


Figure 2.19 Illustration of AlexNet architecture. Source: [24].

2.4 Face Recognition

Face is the physiological characteristic that is most used by humans to identify an individual. When we talk about face recognition we refer to all the techniques that allow a machine to process the image of a face and recognise it. One of the pioneers

in this field was Woodrow Bledsoe, who used a technique to classify photographs of faces digitalised by hand [9]. He also reported some difficulties of this tasks, including the "variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc." [8]. Today, due to the progress in machine learning, more advanced techniques have been adopted, and we are going to discuss some of them in the following sections. Before, we need to introduce the standard pipeline used for face recognition in order to understand how to apply these methods.

A face recognition system allows to perform the three processes that are common to a generic biometric recognition system [26] (Figure 2.20):

- Enrolment. It is the procedure of the acquisition of a biometric trait, also known as sample, (face in our case) that after some stages, in order *preprocessing*, *feature extraction*, and *template generation*, is stored in a database with a unique identifier.

- Verification. In the verification process one sample is compared with a specific template in the database. This comparison is called *matching*, and it produces a similarity score. If this score is higher than a certain threshold, the two traits are considered as belonging to the same person.

- Identification. In case of identification, the acquired trait is compared with all the elements of the database in order to get the identity which the sample belongs to, based on the highest similarity score.

The evaluation of a recognition system is generally performed by estimating the errors done by the matcher. Two kinds of errors are possible:

- Two samples from different people are considered as belonging to the same person (*false match*).

- Two samples from the same person are considered as belonging to different people (*false non-match*).

As we said before, a matcher produces a similarity score as output that is compared with a system threshold. As a consequence, we can have different results based on the value of the threshold. We can evaluate the system using two curves: the *False Match Rate* (*FMR*), also known as *False Acceptance Rate* (*FAR*), or the probability that a false match occurs, and the *False Non-Match Rate* (*FNMR*), also known as *False Rejection Rate* (*FRR*), or the probability that a false non-match occurs. The error rate for which the FMR equals the FNMR is known as *Equal-Error Rate* (*EER*) (Figure 2.21). Based on the application, one wants to have a low FMR (e.g. access control), or a low FNMR (e.g. forensics) [26].

Sometimes the performance of the system is evaluated using the Receiver Operating Characteristics (*ROC*), defined as the plot of the *Genuine Acceptance Rate* ($GAR = 1 - \text{FNMR}$), against the FMR.

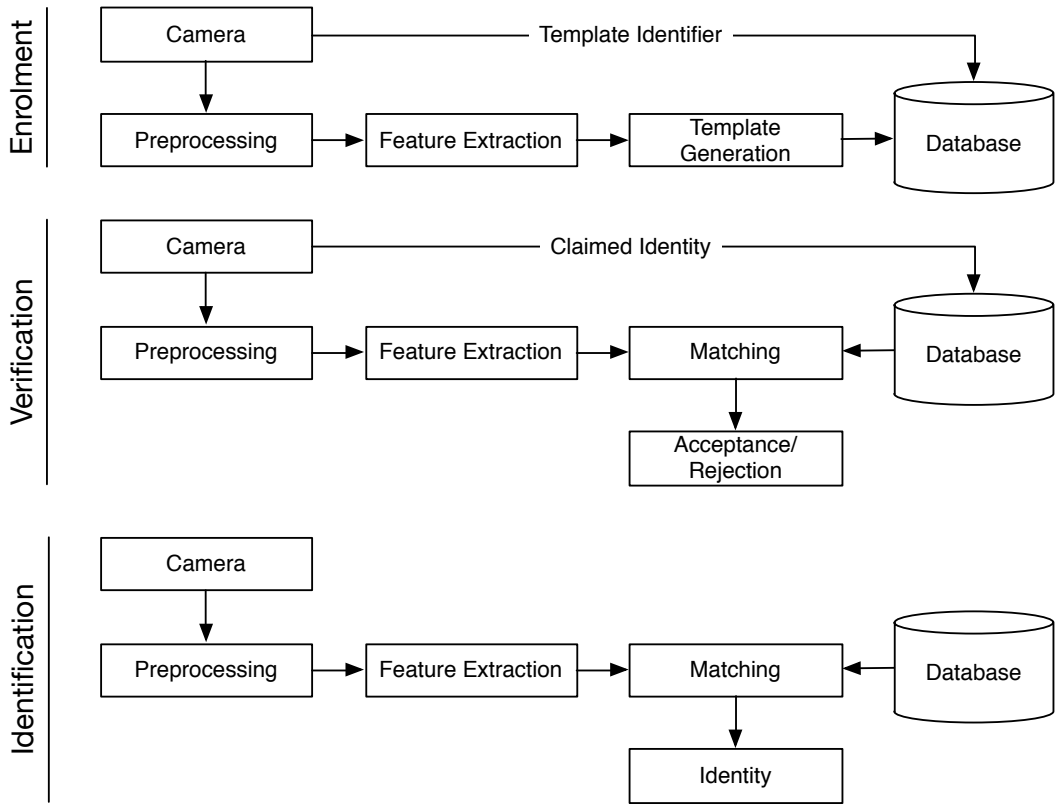


Figure 2.20 Enrolment, verification, and identification for a biometric recognition system.

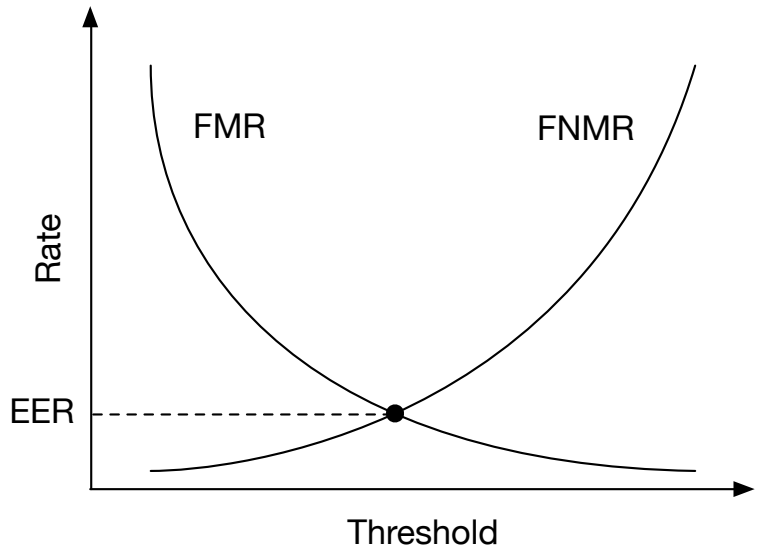


Figure 2.21 False Match Rate, False Non-Match Rate, and Equal Error Rate.

2.4.1 Preprocessing

The preprocessing step is used to improve the recognition accuracy by making the feature extraction easier and more effective. Different approaches can be used, based on the algorithms adopted in the following steps, but generally they deal with noise, lighting conditions and head pose.

Some image enhancement techniques that can be adopted consist of the manipulation of the image histogram, which is a representation of the intensity distribution of the image [15]. In particular, histogram equalisation is a way to normalise the histogram of an image so that it can be more effectively compared with another image acquired under different conditions (Figure 2.22). This technique maps one distribution to another one, which is more uniform, so that ideally all the intensity values are equally probable. To achieve this goal the mapping function should be based on the cumulative distribution function [15]. In particular, if we have a $w \times h$ image I, with an histogram $H(i)$, its cumulative distribution is:

$$H'(i) = \sum_{0 \leq j < i} H(j).$$

In order to use it as a mapping function, its output should be in the range $[0, 255]$:

$$H''(i) = \frac{255}{w \cdot h} H'(i).$$

Then we can use this function to produce the output image:

$$O(x, y) = H''(I(x, y)).$$

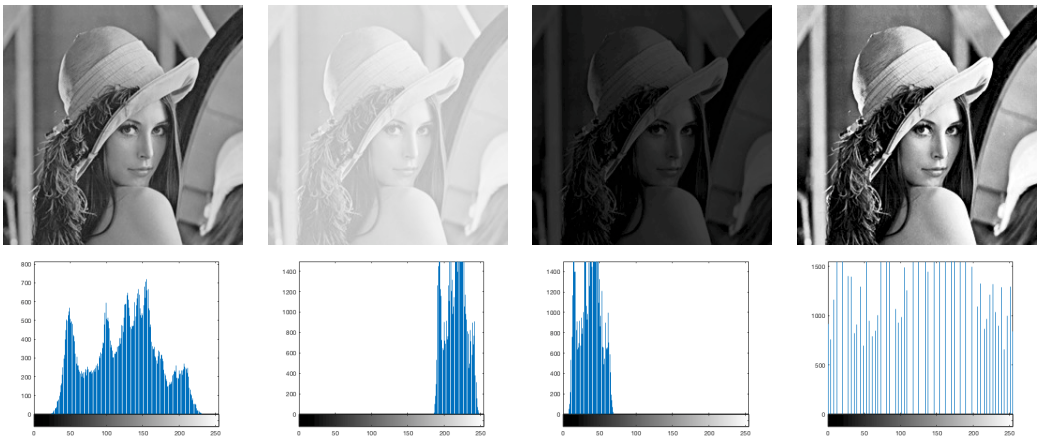


Figure 2.22 Histogram equalisation. The first three images on the top left represent the same image with different intensity distributions. All the three images after the histogram equalisation are really similar (last image). The respective histograms of the images are shown below them.

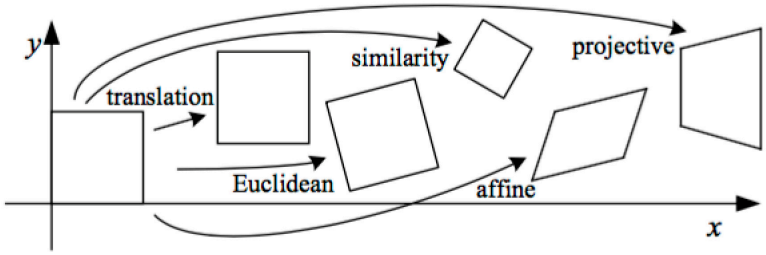
Besides image enhancement techniques, face alignment is a step generally performed in order to obtain a normalised face image. A simple way to achieve this is by applying an image warping based on some landmark points. If we consider 2D linear transformations, they can be obtained by multiplying a 3×3 transformation matrix by the points of the original image in homogeneous coordinates:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = T \begin{pmatrix} x \\ y \\ w \end{pmatrix}.$$

We can define five transformations (Figure 2.23): translation, rigid Euclidean, similarity, affine, and projective. Each of them is represented by a matrix with a certain number of degrees of freedom. Since a point in homogeneous coordinates satisfies the following identity:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = k \begin{pmatrix} x \\ y \\ w \end{pmatrix},$$

then it has 2 degrees of freedom (DOF).



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\, \boldsymbol{I} \mid \boldsymbol{t} \,\right]_{2\times3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\, \boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times3}$	3	lengths $+\cdots$	
similarity	$\left[\, s\boldsymbol{R} \mid \boldsymbol{t} \,\right]_{2\times3}$	4	angles $+\cdots$	
affine	$\left[\, \boldsymbol{A} \,\right]_{2\times3}$	6	parallelism $+\cdots$	
projective	$\left[\, \tilde{\boldsymbol{H}} \,\right]_{3\times3}$	8	straight lines	

Figure 2.23 2D coordinate transformations. Source: [32].

For example, if we want to get a particular affine transformation, which is determined by 6 DOF as shown in Figure 2.23, then a set of 3 points in two images is needed. This is the case of a simple face alignment based on three landmark points: the centres of the eyes and of the mouth (Figure 2.24).

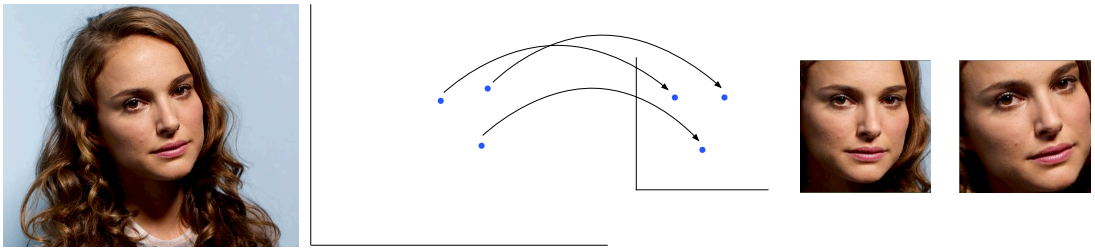


Figure 2.24 Face alignment using an affine transformation determined by three landmark points. The last image represents a crop of the face without alignment.

2.4.2 Feature Extraction

In order to solve the problem of face recognition, it is necessary to deal with high dimensional data. Some techniques can be used to reduce the image dimensions and consider the most important features.

Eigenfaces

Eigenfaces [33] is a method based on principal component analysis (PCA), which is a dimensionality reduction technique with data loss whose purpose is to project data onto the direction in the data space which has highest variance.

In particular, if we have a dataset of observations $\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_N$, with $\mathbf{x}_i \in R^D$, we want to project the data onto a space with a dimensionality $M < D$ maximising the variance of the projected data. We can choose a unit vector $\mathbf{w}_1 \in R^D$ such that $\mathbf{w}_1^T\mathbf{w}_1 = 1$. The data projected onto the direction of $\mathbf{w}_1$ has a mean of $\mathbf{w}_1^T\mathbf{m}$, where $\mathbf{m} = \frac{1}{N}\sum_i \mathbf{x}_i$, and a variance of $\mathbf{w}_1^T S\mathbf{w}_1$, where $S = \frac{1}{N}\sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$ is the data covariance matrix. In order to maximise the variance with respect to $\mathbf{w}_1$ with the constraint $\mathbf{w}_1^T\mathbf{w}_1 = 1$, we have to make an unconstrained maximisation of $\mathbf{w}_1^T S\mathbf{w}_1 + \lambda_1(1 - \mathbf{w}_1^T\mathbf{w}_1)$, where λ_1 is a Lagrange multiplier. If we set the derivative with respect to $\mathbf{w}_1$ equal to zero, then $S\mathbf{w}_1 = \lambda_1\mathbf{w}_1$, which means that $\mathbf{w}_1$ is an eigenvector of S. Since the variance is $\mathbf{w}_1^T S\mathbf{w}_1 = \lambda_1$, then the variance is maximised when $\mathbf{w}_1$, known as principal component, is the eigenvector with the largest eigenvalue. Other principal components can be found by choosing the orthogonal directions that maximise the projected variance. The projection of an observation $\mathbf{x}_i$ onto the PCA subspace is:

$$\mathbf{z}_i = \mathbf{W}^T(\mathbf{x}_i - \mathbf{m}),$$

where the columns of $\mathbf{W}$ are the eigenvectors of S. The reconstruction is given by:

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i + \mathbf{m}.$$

Fisherfaces

PCA finds a data representation that maximises the variance of the data. This does not guarantee that the representation is the best for classification, because the directions discarded by PCA might have discriminative information. For this reason,

linear discriminant analysis (LDA), introduced by Fisher [19] to classify flowers, aims at finding a subspace where the projected observations belonging to different classes are well separated.

In particular, for a two-class (C_1, C_2) problem we want to find the unit vector $\mathbf{w}$ that maximises:

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2},$$

with

$$m_1 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t r^t}{\sum_t r^t}, \qquad m_2 = \frac{\sum_t \mathbf{w}^T \mathbf{x}^t (1 - r^t)}{\sum_t (1 - r^t)}$$

and

$$s_1^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_1)^2 r^t, \qquad s_2^2 = \sum_t (\mathbf{w}^T \mathbf{x}^t - m_2)^2 (1 - r^t)$$

where $r^t = 0$ if $\mathbf{x}^t \in C_2$ and $r^t = 1$ if $\mathbf{x}^t \in C_1$ [5].

Figure 2.25 shows an example of dimensionality reduction where the data is projected onto the direction of the principal component and of the vector found by LDA. It is possible to see that the projection onto the direction of the principal component causes an overlap of the two classes.

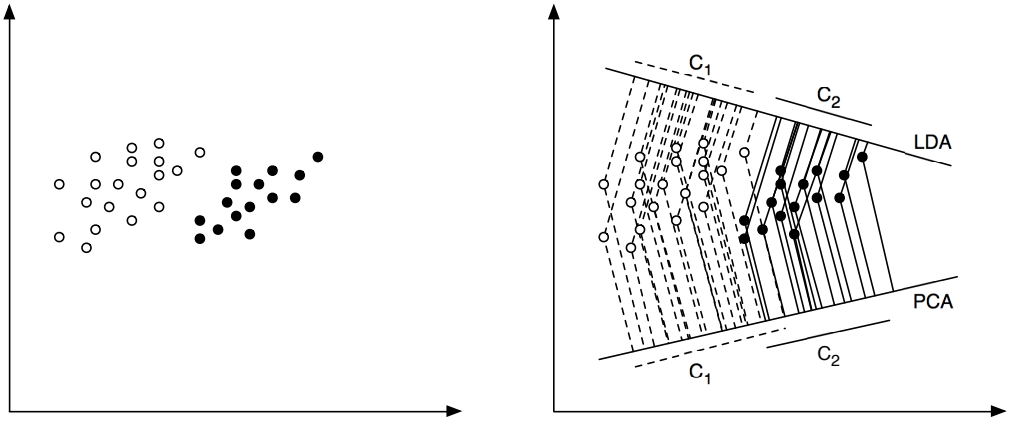


Figure 2.25 Difference between PCA and LDA for a two-dimensional two-class problem.

Local Binary Patterns Histograms

While Eigenfaces and Fisherfaces consider an image as a vector in a high-dimensional space, Local Binary Patterns Histograms (LBPH) [4] focuses on local features. This technique consists of dividing the image into squares and consider every 3×3 window in the squares. The intensity of each pixel in the window is compared with the intensity of the central pixel: if it is greater a value of 1 is assigned to the pixel, 0 otherwise. In this way, we got an 8-bit binary code for each window (Figure 2.26). Then, a histogram is built for each square and all the histograms are concatenated.

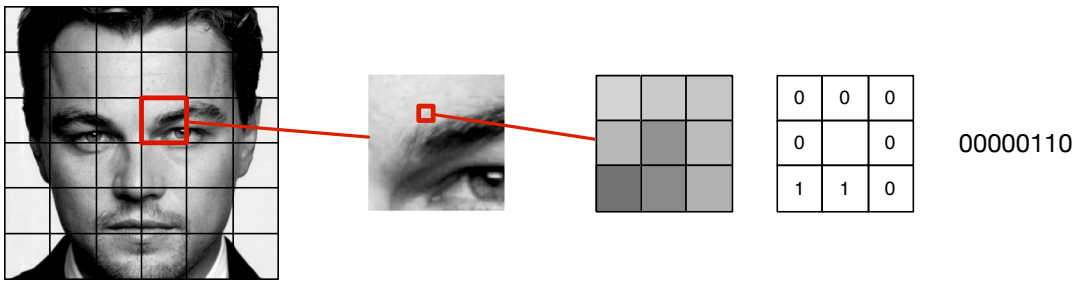


Figure 2.26 LBP code.

Convolutional Neural Networks

Convolutional Neural Networks are the current state of the art for face recognition. A CNN can be seen as a feed-forward network consisting of different layers (Figure 2.27). The most important ones are:

- Input layer. It is the image we need to process.
- Convolutional layer. It performs a convolution between the previous layer and some learnt filters.
- Activation layer. It applies a non-linear activation function f. The most common functions are [2]:
  - Binary step. Even though it is the simplest form of activation function, it is not used because it is not differentiable at 0, making a gradient-based algorithm impractical.

$$f(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

  - Sigmoid. It has a behaviour similar to the binary step, allowing the transmission of a signal if it is bigger than a threshold, without its main drawback. It is rarely used in CNNs because it has two saturation regions for large positive/negative input, known for 'killing' the gradient and rising the training time [24].

$$f(x) = \frac{1}{1 + e^{-x}}.$$

  - Tanh. Similar to the non-linear sigmoid function, but the output is zero-centred.

$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1.$$

  - ReLU. The Rectified Linear Unit is the most popular activation function. It does not present a saturation region for positive input, and it allows to get an improvement in convergence of six times if compared to the tanh unit [24].

$$f(x) = \max(0, x).$$

- Pooling layer. It reduces the size of the representation and guarantees robustness to small transformations. The process consists of taking small regions of the previous layer and extract the mean or the maximum value (max-pooling).
- Fully-connected layer. It performs a linear combination with learnt weights of all the elements of the previous layer.
- Output. It consists of a number of units that equals the number of categories of the classification problem.

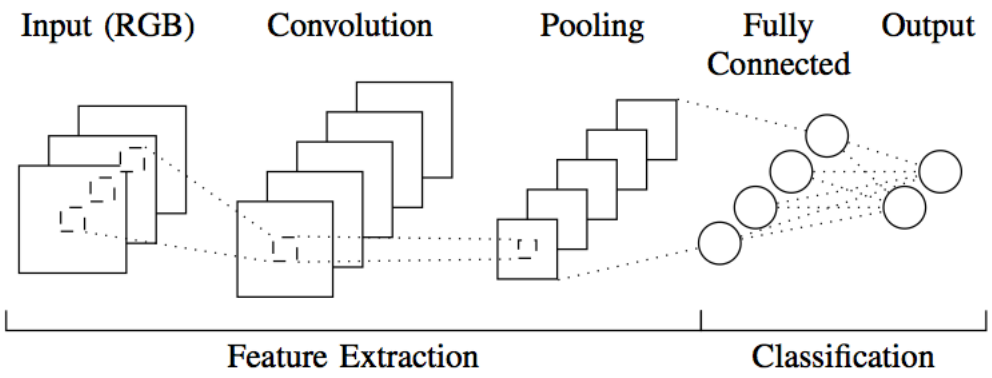


Figure 2.27 Simple CNN. Source: [21].

In summary convolutional neural network performs both feature extraction and classification. If we are only interested in learning a representation of the data, we can take a pretrained network and use the output of the second last layer as features to train a linear classifier, like a support vector machine.

Many researches have been done to adopt CNNs for face recognition. For example Parkhi et al. [27] trained a deep architecture of 37 layers obtaining high performance on some benchmark datasets, like the Labeled Faces in the Wild [22]. The problem of a deep network is that a huge amount of parameters need to be learnt during training, and this require a large database. Also, the real-time performance of an embedded system can be low. For this reason Wu [37] used a shallower network able to achieve high recognition performance with aligned faces.

2.4.3 Matching

We explained before the difference between identification and verification. Based on the kind of problem we want to solve, we have different approaches. Generally before matching, the extracted features are normalised, in order to avoid that the final result is biased by the different ranges of values that the features can assume. We can use the L2-norm for this purpose:

$$|\mathbf{x}|_2 = \sqrt{\sum_{k=1}^{n} |x_k|^2} \quad \text{where} \quad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}.$$

In case of identification, where we want to determine which individual among a list of identities in a database a sample belongs to, we can think to use a classifier, like k-nearest neighbours (k-NN) or support vector machine.

k-NN is one of the simplest classifier where a test image is assigned to the most frequent class of the k closest feature vectors extracted from the training images.

On the other hand, SVM is a binary classifier which learns the decision boundary that maximises the margin between the classes (Figure 2.28). The decision function can be defined as $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i \; y_i \; \mathbf{x}_i \cdot \mathbf{x} + b$, where $\mathbf{w}$ is the normal vector to the decision boundary, y_i can be 1 or -1 indicating the class to which $\mathbf{x}_i$ belongs, and $\alpha_i > 0$ for the $\mathbf{x}_i$ lying on the margin (known as support vectors) and $\alpha_i = 0$ for the others. In case of non-linearly separable data it is possible to apply the kernel trick where a kernel function is used instead of the dot product. The basic kernels are the following ones [16]:

- Linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$.
- Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \; \mathbf{x}_i^T \mathbf{x}_j + r)^d, \gamma > 0$.
- Radial basis function: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2}, \gamma > 0$.
- Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \; \mathbf{x}_i^T \mathbf{x}_j + r)$.

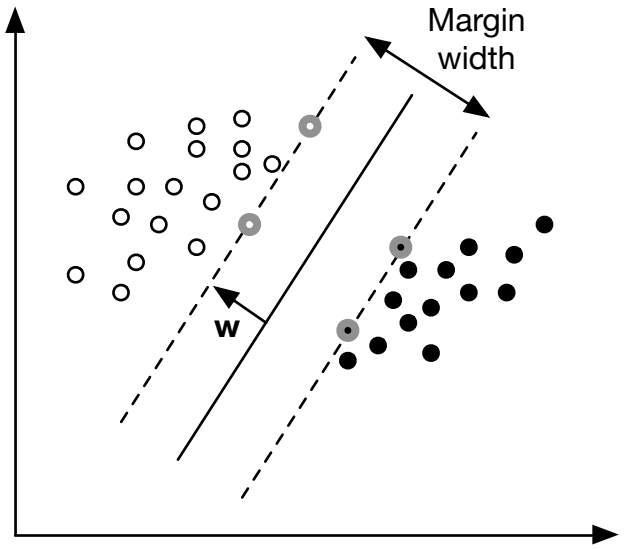


Figure 2.28 Support vector machine.

Regarding verification, the goal is to determine whether the representations (or sets of features) of two samples belong to the same identity. The comparison between the two representations is usually made by using the Euclidean distance-based similarity or the cosine similarity, but it is also possible to learn a metric for the specific task. In particular, given two feature vectors, $\mathbf{x}$ and $\mathbf{y}$, the Euclidean distance is defined as:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

This represents the distance between the two vectors, but in order to have a similarity measure in the range $[0, 1]$, we can add 1 to the distance, and invert it [29]:

$$s_E(\mathbf{x}, \mathbf{y}) = \frac{1}{1 + d(\mathbf{x}, \mathbf{y})}.$$

On the other hand, the cosine similarity measures the cosine of the angle between them:

$$s_C(\mathbf{x}, \mathbf{y}) = cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}|_2 |\mathbf{y}|_2}.$$

Figure 2.29 shows the Euclidean distance and the angle between two 2D vectors.

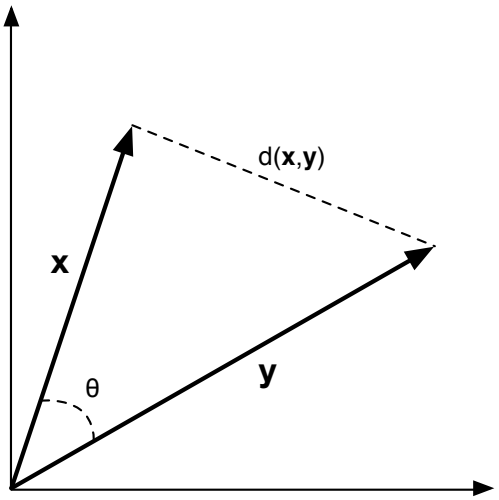


Figure 2.29 Euclidean distance and angle between two 2D vectors.

It is also possible to solve the identification problem, by repeating the approach used for verification multiple times. More specifically, we can compute the similarity score between the given sample and all the elements in the database. The output in this case will be the identity with the highest score.

2.5 Speaker Recognition

The stages needed to recognise a person through voice are similar to the ones in a face recognition system, with the approaches that differ because of the different modality. In this work we consider a text-independent system based on i-vectors.

If we have a discrete signal that represents the voice of a person, the first step in the process is called labelling. The idea here is to divide the track into segments that have relevant information, i.e. speech. The most common way to do so is

to build a model of the speech energy and apply a threshold in order to label the segments lower than a certain level as "silence". It is also possible to perform this step after feature extraction in order to consider only the features of the speech segments.

A robust representation of speech for voice recognition is the Mel-frequency Cepstrum (MFC), made by Mel-frequency Cepstral Coefficients (MFCCs), that is a model based on human hearing perception. First, the audio segments are divided into overlapping frames of N samples each, and they are windowed, generally using Hamming window, to avoid signal discontinuities at the start and at the end of the frame, which generate high-frequency components in the Fourier transform of the signal. After that, the Discrete Fourier Transform (DFT) is computed, and this allows us to operate in frequency domain. Pre-emphasis is performed "for flattening the magnitude spectrum and balancing the high and low frequency components" [25], the obtained signal is filtered with a filter bank in order to have a higher resolution at low frequency, and a log function is applied to have a better response, as the perception in the human hearing system. The final step is to calculate the MFCCs using the Discrete Cosine Transform (DCT). The concatenation of these coefficients will be the feature vector used for recognition, and it is generally normalised using the mean-variance normalisation.

Before going on with the other steps, we need to introduce the concept of Universal Background Model (UBM). It is defined as a model "to represent general, person-independent feature characteristics to be compared against a model of person-specific feature characteristics when making an accept or reject decision" [28]. The UBM is modelled as a mixture of Gaussians (GMM) and can be seen as

$$p(\mathbf{x}) = \sum_i p(\mathbf{x}|G_i)P(G_i)$$

where $p(G_i)$ are the mixture weights (priors) and $p(\mathbf{x}|G_i)$ are the component densities. Each component is a Gaussian, determined by the mean (μ_i) and the covariance (S_i) parameters. Then, in order to find the GMM, we need to estimate the parameters $\Phi = \{P(G_i), \mu_i, S_i\}$. This can be done by training a large database with the Expectation-Maximisation (EM) algorithm, which is an iterative algorithm consisting of two steps: in the E-step "the posterior probability that each Gaussian generates each datapoint" [1] is computed; in the M-step the parameters of each Gaussian is changed "to maximise the probability that it would generate the data it is currently responsible for" [1].

At this point, it is possible to represent a speaker utterance with a supervector, which is built by stacking the means of the GMM components, as it follows:

$$\mathbf{M} = \mathbf{m} + T\mathbf{w},$$

where $\mathbf{m}$ is the UBM supervector, T is a matrix called total variability matrix, and $\mathbf{w}$ is called identity vector, or i-vector [14].

Given two i-vectors extracted from two audio samples, we can see if they belong to the same identity by computing the cosine similarity, as seen for face recognition. Figure 2.30 shows the whole pipeline of the system.

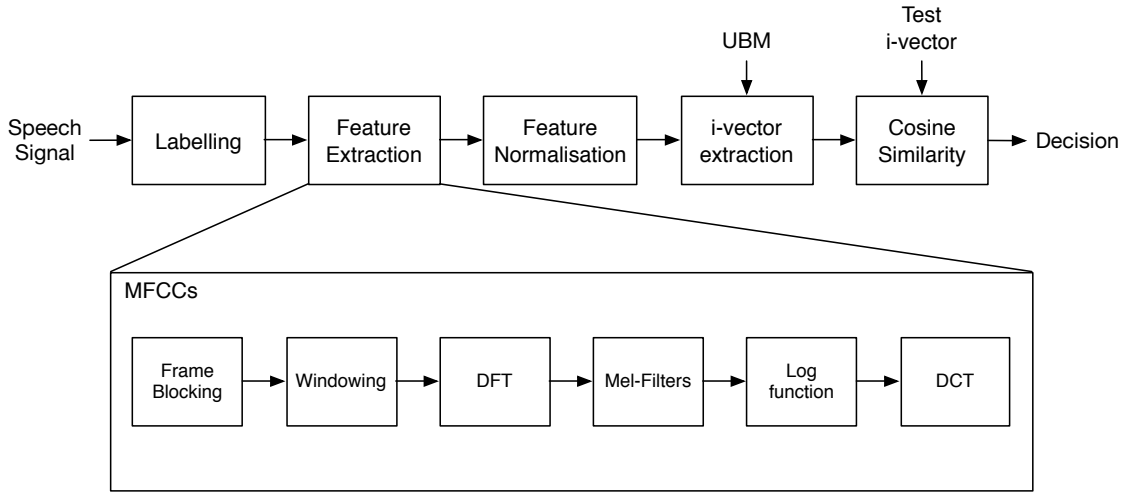


Figure 2.30 Pipeline of a speaker recognition system based on i-vectors.

2.6 Multimodal Biometric Fusion

If a system allows to use more sensors (like a camera and a microphone), it is possible to combine the information from two modalities and build a multimodal system. In this way, the strengths of one modality compensates the weak points of the others. In our case, focusing only on face recognition can be a problem in some situations, for example when a user is watching the TV in low lighting conditions, if the face is partially occluded, or when the head-pose of the person is too different from the samples in the database. Other issues can arise when a speaker-recognition only system is adopted (e.g. the room can be too noisy, the user utterances can be too small). As a consequence, robustness can be gained by fusing the inputs of the two modalities.

There are two main techniques to perform a fusion of multiple modalities (Figure 2.31):

- Early fusion. The features extracted from the unimodal streams of information are combined in order to get a multimodal representation that is used in the final learning method.
- Late fusion. Models from different modalities are learned independently, and the obtained scores are combined in the end.

The second approach is applied most often for multimodal recognition systems. There are three steps in the fusion process [17]:

- Score normalisation. It is used in order to transform the scores for the different modalities to values that varies in a common range. The simplest technique is the min-max normalisation where the normalised score of the single modality is calculated as it follows:

$$s^* = \frac{s - \min}{\max - \min}.$$

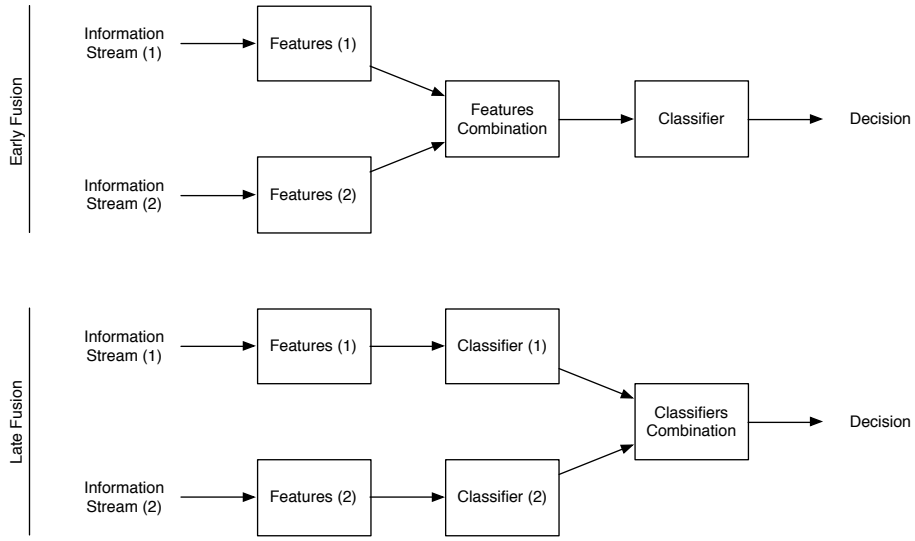


Figure 2.31 Early and late fusion approaches.

Other techniques can be used, like the z-score, where the normalised score is given by:

$$s^* = \frac{s - \mu}{\sigma},$$

with μ and σ that represent the mean and the standard deviation of the data respectively.

- Modality weighting. The purpose of this step is to allow the modality with better performance to have a greater impact on the final score. It is possible to adopt adaptive modality weighting schemes [17] or a fixed modality weighting scheme, where the score of each modality is weighted based on their identification rate.

- Modality combination. This is the stage where the unimodal scores are combined together. Some methods can be found in the work of Snelick et al. [30] and are shown in Table 2.1.

Simple sum	$\sum_i s_i^*$
Minimum score	$\min_i(s_i^*)$
Maximum score	$\max_i(s_i^*)$
Sum of probabilities	$\sum_i P(genuine\|s_i^*)$
Product of probabilities	$\prod_i P(genuine\|s_i^*)$

Table 2.1 Modality combination techniques. $P(genuine|s_i^*)$ is the posterior probability of the *genuine* match given s_i^*. Source: [30].

2.7 Implementation Details

As seen in Section 2.1.1, the final prototype consists of some modules to perform the recognition of the user, the eye tracking, and to show the graphical interface. Figure 2.32 shows the main blocks of the system: a HD camera captures the video stream that is processed with a laptop, and the GUI is transferred to a B&O TV via HDMI. The whole prototype has been implemented in C++ on a Windows platform, but no platform specific libraries have been used, making a porting to Linux easy to obtain.

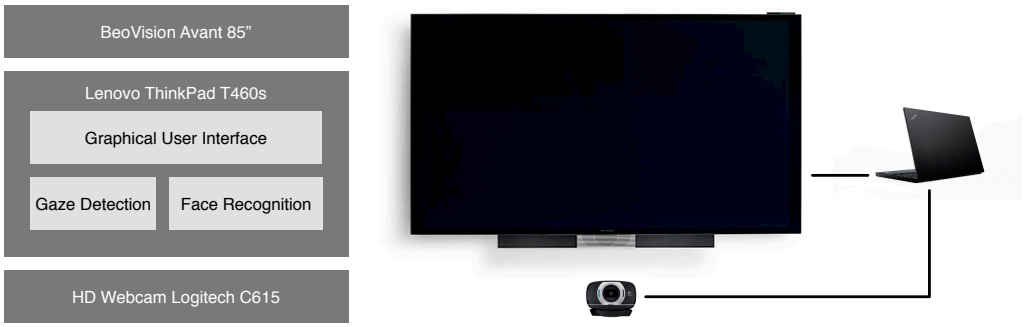


Figure 2.32 Modules implemented in the final prototype along with the used hardware.

Even though the speaker recognition has not been integrated in the prototype, we implemented it to check the performance of a multimodal system for a future integration.

2.7.1 Gaze Detection

This module is not part of this work, but we briefly describe it. The library used for gaze tracking is OpenFace, an open source facial behaviour analysis toolkit [6] [7] [36]. With this library the face is detected adopting a HOG algorithm, and three tridimensional vectors, two for the directions of the eyes ($\mathbf{v}_{e1}$, $\mathbf{v}_{e2}$) and one for the head-pose ($\mathbf{v}_h$), are extracted. We combined the two vectors $\mathbf{v}_{e1}$ and $\mathbf{v}_{e2}$ together and normalised the final vector:

$$\mathbf{v}_e = \frac{\mathbf{v}_{e1} + \mathbf{v}_{e2}}{|\mathbf{v}_{e1} + \mathbf{v}_{e2}|_2}.$$

Then the vector that represents the direction of the gaze ($\mathbf{v}_e$) and the one that represents the head-pose ($\mathbf{v}_h$) have been used as features of a SVM classifier (included in the OpenCV 3.1 library [11]) whose output is an integer between 0 and 3 that is associated with one of the corners of the TV screen.

2.7.2 Face Recognition

The face recognition module represents the core of the whole work, for this reason we are going to provide more details about it. The two libraries used are OpenCV (with the contrib extra-modules) and OpenFace.

The implemented module includes basic techniques (e.g. eigenfaces) and advanced ones (e.g. deep learning). It is easy to modify to include more functionalities in the future, due to its simple structure that consists of two classes, one struct, and two enumerations, as shown in Figure 2.33.

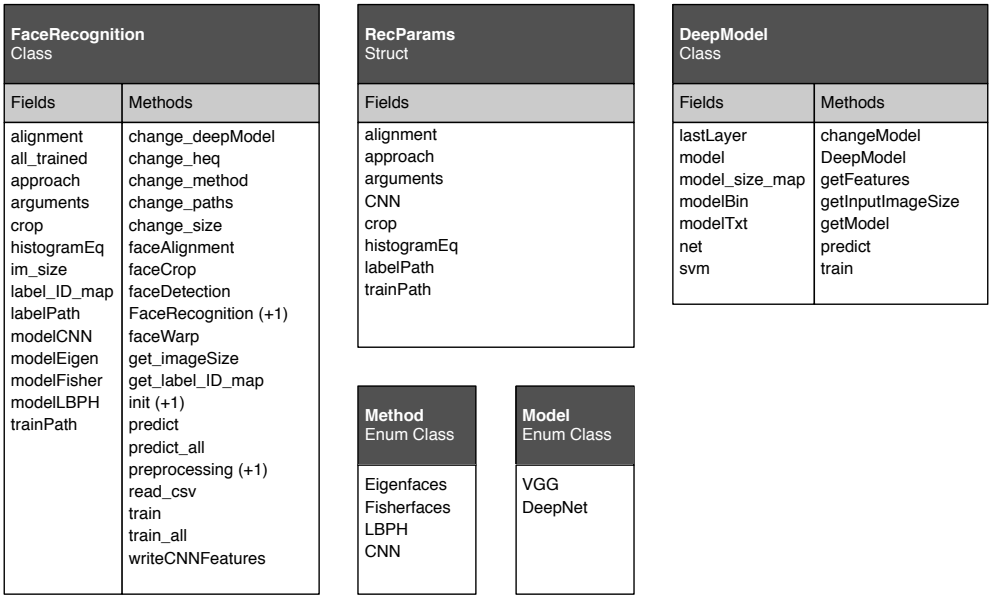


Figure 2.33 Structure of the face recognition module.

The main class is FaceRecognition, where we have the methods to preprocess the image, train the classifier and make the prediction. When the class is used, various parameters can be set with the RecParams struct, like the preprocessing techniques (face crop, face alignment, histogram equalisation), the approach (Eigenfaces, Fisherfaces, local binary pattern histograms, and convolutional neural networks, as listed in the Method enumeration), the path of the training and the label files, and the CNN model (that can be the VGG deep face [27] or the light CNN [37], as listed in the Model enumeration). When Eigenfaces, Fisherfaces or LBPH are used, the class provides a wrapper of the OpenCV methods, which adopts a nearest neighbour classifier. On the other hand, the CNNs are used as feature extractors whose second-last layer output (a 4096-dimensional vector for VGG-Face and a 256-dimensional vector for the light model) is used as features of a SVM classifier with a radial basis function as kernel. In order to handle the CNN, a class named DeepModel has also been implemented.

The module can be used as it follows:

```
int main(int argc, char **argv)
{

    // Put the arguments in a std::vector
    // ...

    // Set the parameters for the face recognition system
    RecParams faceRecParams;
    faceRecParams.arguments = arguments;
    faceRecParams.approach = Method::CNN;
    faceRecParams.CNN = Model::VGG;
    faceRecParams.crop = false;
```

```
    faceRecParams.alignment = false;
    faceRecParams.histogramEq = 0;
    faceRecParams.labelPath = "at_label.txt";
    faceRecParams.trainPath = "at_training.txt";

    // Create a FaceRecognition instance
    FaceRecognition recognizer(faceRecParams);

    // Train the system
    recognizer.train();

    // Get the map between IDs and names
    std::map<int, std::string> label_ID_map(recognizer.get_label_ID_map());

    // Read the test image
    //

    // Make the prediction (prediction stores the ID)
    int prediction = recognizer.prediction(test_image);

    // Get the name of the user
    std::string user_name = label_ID_map[predition];

    return 0;

}
```

2.7.3 Graphical User Interface

We implemented a state machine following the consideration explained in Section 2.1.1. The program can be in one of four states (or modes): single picture mode, gaze detection mode, PiP mode, and sound selection mode. Based on the state, different information appear. All the animations have been implemented in OpenCV by displaying an image or a video on top of a background video. Figure 2.34 illustrates how the information are displayed if the user look at one of the corner for 3 seconds. If the user keeps looking at the corner that shows the favourite channel for other 3 seconds, a PIP will appear.

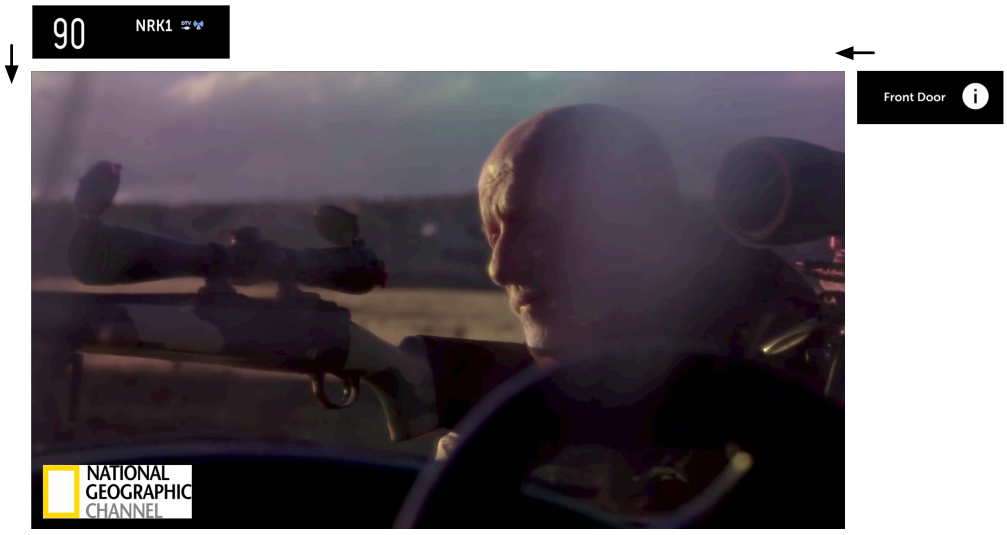


Figure 2.34 Graphical user interface.

2.7.4 Speaker Recognition

For speaker recognition we used the speech signal processing toolkit (SPro) [20] for feature extraction and the ALIZE / LIA_RAL speaker verification library [10] for the other stages of the system.

First, we extracted the MFCCs features of all the audio files using the sfbcep (filter-bank cepstral features) tool of the SPro library. Among the options that can be controlled, we:

- Activated the Mel frequency warping to space the filters along the Mel frequency scale (-m).
- Set the number of filters to 24 and the number of the output coefficient to 19 (-n 24 -p19).
- Added the first and second derivative of the coefficients to the output vector (-D -A).

Then, the EnergyDetector program creates a label file that indicates the speech segments, and the features are normalised with the NormFeat program. The following step is the training of the UBM with the TrainWorld program. The two main parameters that has been set are:

- The number of distributions (mixtureDistribCount) to 512.
- The number of training iterations (nbTrainIt) to 25.

After that, the total variability matrix is estimated using the TotalVariability program and the i-vectors are extracted with the IvExtractor one. We set the rank of the total variability matrix (totalVariabilityNumber), which is the dimension of the vector space generated by its columns/rows, to 40. Finally, the cosine similarity score is computed.

2.7.5 Multimodal Fusion

The fusion of face and voice has been implemented in MATLAB. Even if cosine similarity has been used for both the modalities, the scores varies in different ranges. This is because the coordinates of the feature vectors extracted for face recognition are non-negative (due to the ReLU layer at the end), making the outcome bounded in $[0, 1]$. On the other hand, for speaker recognition the score is between -1 and 1. Then, the unimodal scores have been normalised using the min-max normalisation and the results are combined with a weighted sum of scores:

$$s_{final} = a s^*_{face} + (1 - a) s^*_{voice}$$

where a and $1 - a$ are the weights of the two modalities.

2.8 Results and Discussion

For our experiments on face recognition we used the AT&T face database [12]. It consists of ten images of each of 40 subjects. The images have been taken in different conditions (varying the lighting, facial expressions and facial details). Each image is 8-bit grayscale and has a size of 92×112 pixels.

For a first evaluation of the different techniques we decided to solve an identification problem. In particular we divided the data into two sets: a training set, consisting of the first 8 images per identity, and a test set, consisting of the last 2 images per identity. We trained a classifier (1-NN for Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms as used in OpenCV, and SVM for the CNN approaches) on different subsets of the training set, in order to evaluate how the accuracy changes if we vary the size of the training set.

In the first experiment we used Eigenfaces in order to determine how the accuracy varies when we change the number of principal components. As shown in Figure 2.35, if we use fewer than 10 principal components, we have poor results because they are not able to preserve the variability of the data needed for the recognition. On the other hand, choosing more than 80 components guarantees better performance, but the accuracy is still low for a small training set (around 65% when one image per person is used). In the following experiments, we used 120 principal components.

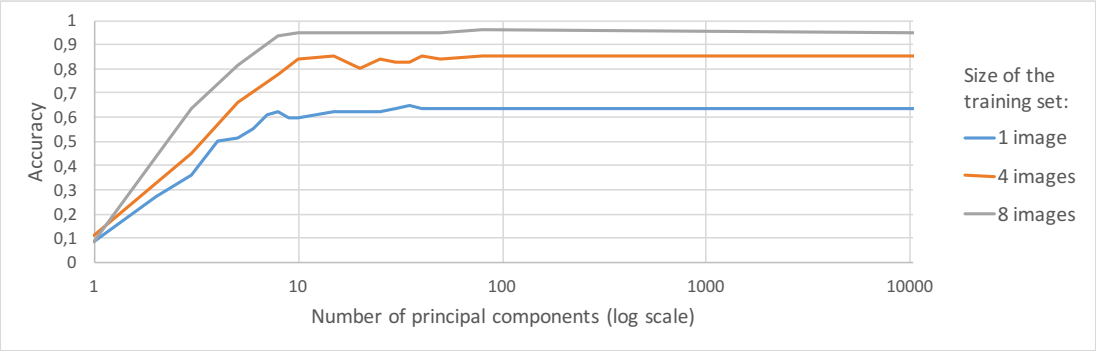


Figure 2.35 Experiments using Eigenfaces.

Then, we compared the different techniques by choosing the first n images of each person (with $1 \leq n \leq 8$) as the training set, and the last two as the testing set. For Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms we used the original images, but for the CNN approaches we needed to make some changes. The pre-trained networks we used accept 3-channel images as input, so we replicated the grayscale image in each channel. Regarding the size, the light CNN needs a 128×128 pixels image, while VGG-Face a 224×224 pixels image, so we resized all the images to these dimensions. As it can be seen in Figure 2.36, the SVM trained on VGG-Face features outperforms all the other methods getting a 100% of accuracy all the times. On the other hand, the other approaches show a similar trend: they present a low accuracy (around 60%) when only one image per person is used for training and the performance gets better when more data is available. The best techniques overall are Eigenfaces and LBPH that reach a 95% of accuracy

with 8 images per person in the training set. Also the Fisherfaces method has a good accuracy (90% or above) when at least 6 images are used for training, while the light CNN representation allows to achieve only around 80% of accuracy.

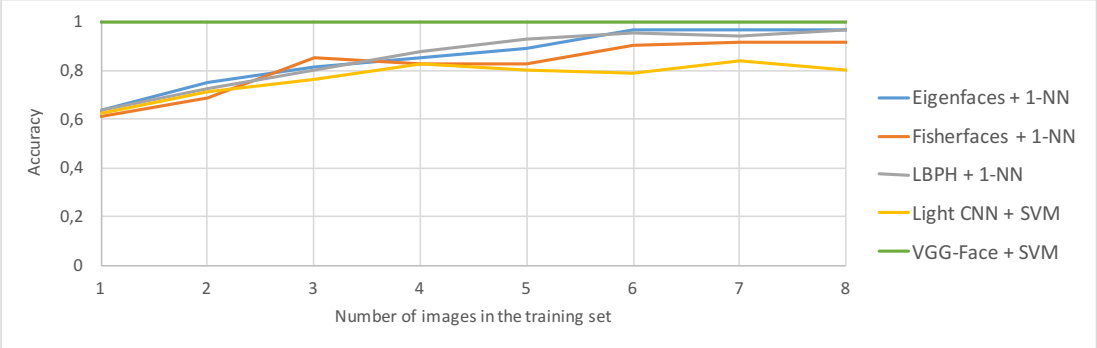


Figure 2.36 Performance of the different approaches without pre-processing.

It is worth saying that these results are obtained without applying any pre-processing techniques. In particular, the light CNN has been trained using aligned faces, and this may justify the low performance. Firstly we tried to equalise the histogram of the images, but this did not make much difference (Figure 2.37).

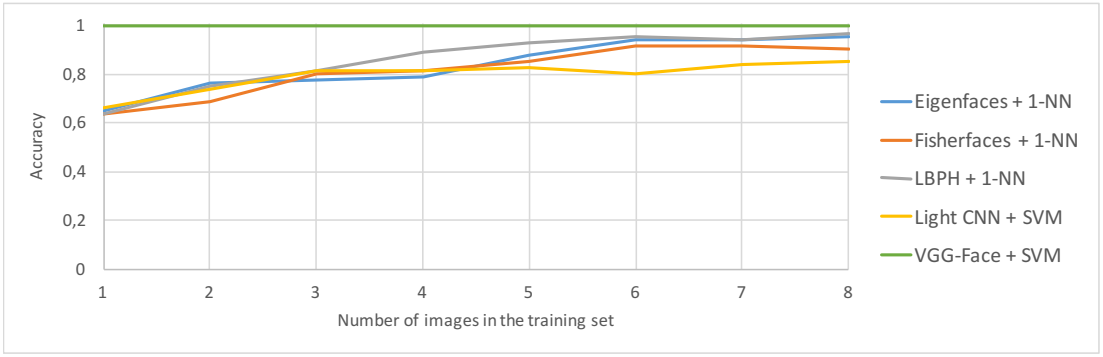


Figure 2.37 Performance of the different approaches after histogram equalisation.

Then, using the functionalities offered by the OpenFace library, we aligned the images based on three keypoints (eyes and mouth) using an affine transformation and cropped them to 224×224 (resized to 128×128 for the light CNN). We can see in Figure 2.38 that the performance of the light CNN increased significantly, reaching an 87.5% and a 98% of accuracy when only one or more than three images are used for training respectively, making it comparable to VGG-Face representation when enough data is available. Also the Fisherfaces method took advantage of the alignment, at least when enough data is available, with an accuracy rate that soared from 47.5% with one image per person in the training set to 100% when at least 7 images per person are in the training set. On the other hand, the performance of the Eigenfaces and the LBPH methods are slightly worse if compared to the previous tests.

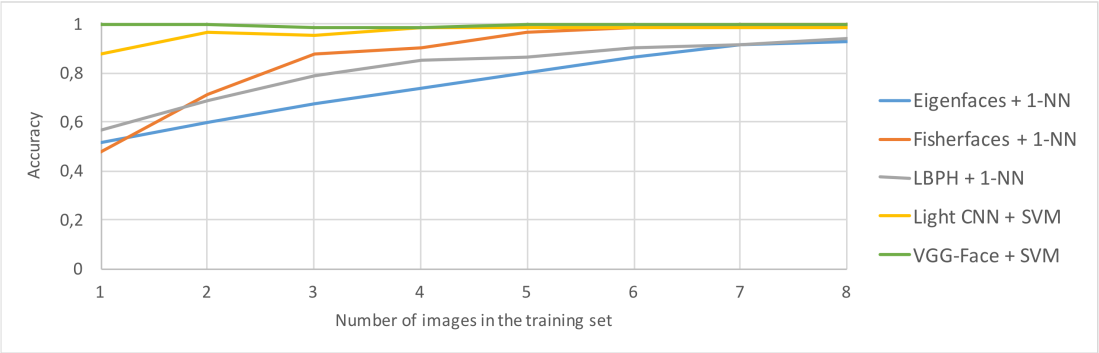


Figure 2.38 Performance of the different approaches after histogram equalisation and face alignment.

Based on these results, it makes sense to choose the feature extraction performed with VGG-Face and investigate more its performance in terms of FMR, FNMR, EER, and ROC on the database. In order to estimate the FNMR, we matched each image in the database with the remaining images of the same person, obtaining a total of $[(10 \cdot 9)/2] \cdot 40 = 1800$ genuine tests. The FMR was computed by matching the first image of each identity in the database with the first image of the other identities, with $(40 \cdot 39)/2 = 780$ impostor tests. In both cases no symmetric matches are performed. The metric used for matching was the cosine similarity. Figure 2.39 shows the FMR and the FNMR curves. The EER is 0.641%, obtained for a value of the threshold of 0.3226.

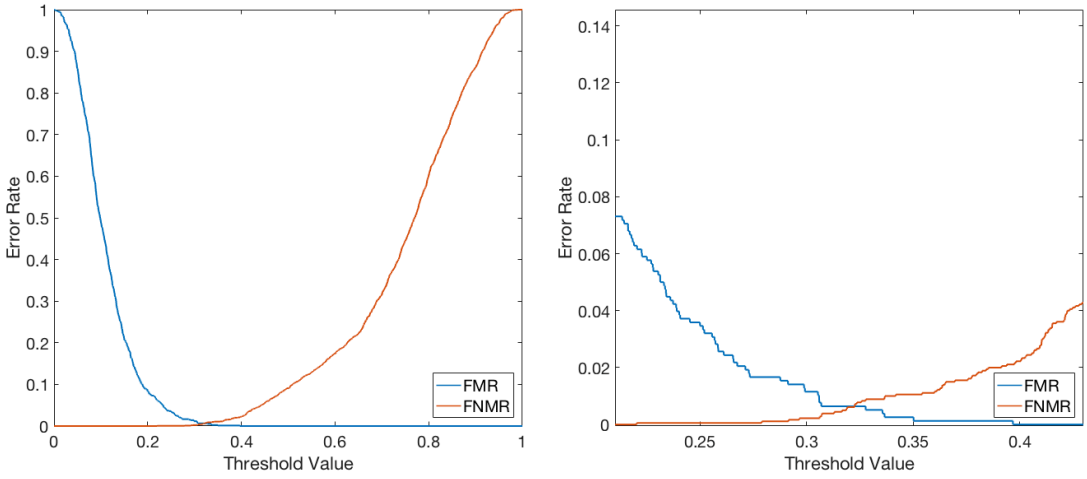


Figure 2.39 FMR and FNMR curves for face recognition. On the right a close-up of the curves around the EER point is shown.

We report also the ROC curve of the system (Figure 2.40).

In order to evaluate the speaker recognition system and make a fusion between the information from speech and face, a multimodal database has been created. We downloaded 20 different TED talks [3] and extracted 10 minutes of speech and 10 images of cropped faces for each talk. Regarding the speech, we trimmed each 10-minute segment into ten parts of one minute each. We also used 3 minutes of speech

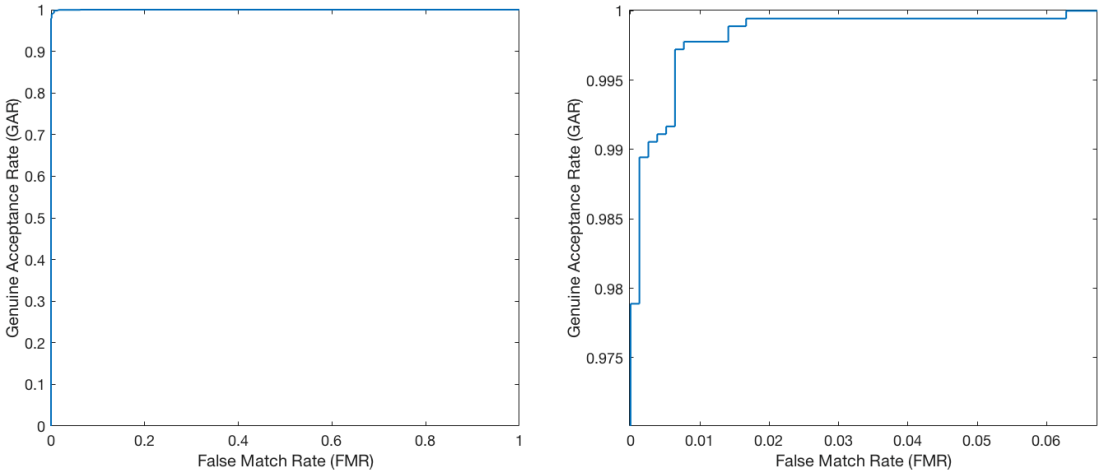


Figure 2.40 ROC curve for face recognition. On the right a close-up of the curve is shown.

from other 63 different talks to train a Universal Background Model. Every segment of speech is a 14-bit mono track with a sample rate of 8 kHz saved in the NIST SPHERE format. On the other hand, we wanted to make the face recognition more challenging, because the recording conditions of the video were good and generally far from a real-case scenario. Then, we reduced the dimensions of the face images from 200×200 to 50×50, changed the intensities and drew a 30×10 black box in different positions to simulate bad lighting conditions and occlusion respectively. Figure 2.41 shows 10 face images of the database before and after the described procedure.

Figure 2.41 Examples of the images obtained from the TED videos. For each couple of images, the top one represents the crop of a frame of the original video, while the bottom one is after the size reduction and the bad lighting conditions or the occlusion simulation.

We split the database into two sets of 10 identities each. The first one is used to see the effect of some parameters changes, while the other one to actually test the system. For face recognition we used the CNN approach, using the VGG-Face model without any preprocessing. For speaker recognition, i-vectors are extracted from the samples, and the cosine similarity is used as matching score. We estimated the FMR, and the FNMR in a similar way as before. We matched each image/utterance in the database with the remaining images/utterances of the same person, obtaining a total of $[(10 \cdot 9)/2] \cdot 10 = 450$ genuine tests. Then, we matched the n^{th} image/utterance of each identity in the database with the n^{th} image/utterance of the other identities, obtaining $[(10 \cdot 9)/2] \cdot 10 = 450$ impostor tests. In both cases no symmetric matches are performed. Figure 2.42 shows the FMR and the FNMR curves for the face recognition and the speaker recognition systems. The EER for the first one is 12.22%, whereas for the second one is 13.33%.

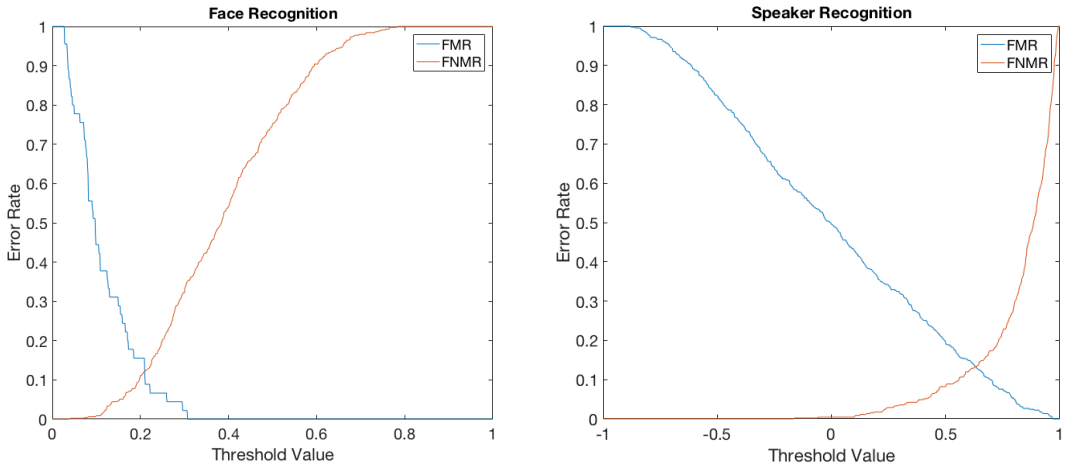


Figure 2.42 FMR and FNMR curves for face recognition and speaker recognition (training set).

At this point, we made a fusion of the two modalities as explained in Section 2.7.5. In Figure 2.43 we can see the effect of the variation of the parameter a on the ROC curve of the system. When we equally weight the two modalities ($a = 0.5$) we obtain the best result, even though the ROC curve plotted for $a = 0.6$ overcomes the other one in some regions. On the other hand, if we decide to prefer one modality ($a = 0.3$ or $a = 0.7$) we experience a significant performance drop. For this reason we chose $a = 0.5$, and, as shown in the right part of Figure 2.43, the multimodal system provides better performance than any of the other two unimodal systems, reaching an EER of 6.22%.

We repeated the experiments in the same conditions as before on the test set. In this case, we can see from Figure 2.44 that the face recognition system performs slightly worse than the speaker recognition one. The EER of the two methods are 16.22% and 9.33%.

When we make a fusion of the two modalities ($a = 0.5$), after a min-max normalisation of the scores, we have a substantial improvement of the performance, obtaining an EER of 3.11%. In Figure 2.45 the FNMR and FMR curves for the fusion method are shown, as well as the three ROC curves.

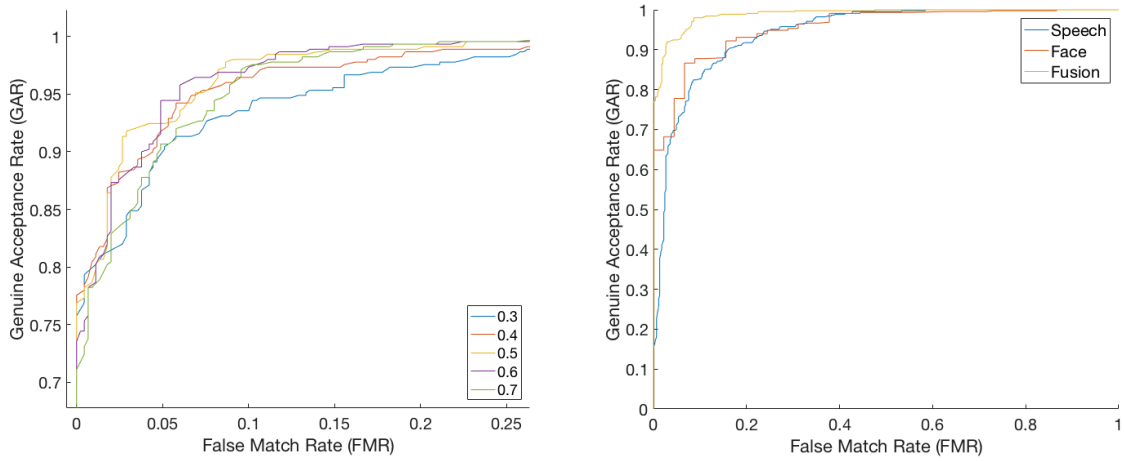


Figure 2.43 ROC curves for the fusion method and the unimodal systems (training set). Left: ROC curves for the fusion method when the a parameter varies between 0.3 and 0.7. Right: ROC curve for the fusion method with $a = 0.5$ compared with the ROC curves for the unimodal systems.

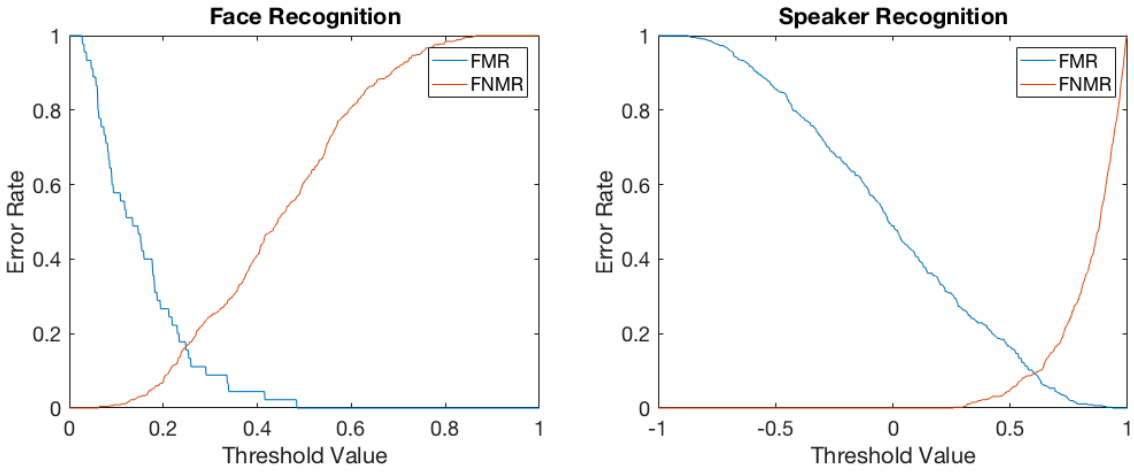


Figure 2.44 FMR and FNMR curves for face recognition and speaker recognition (test set).

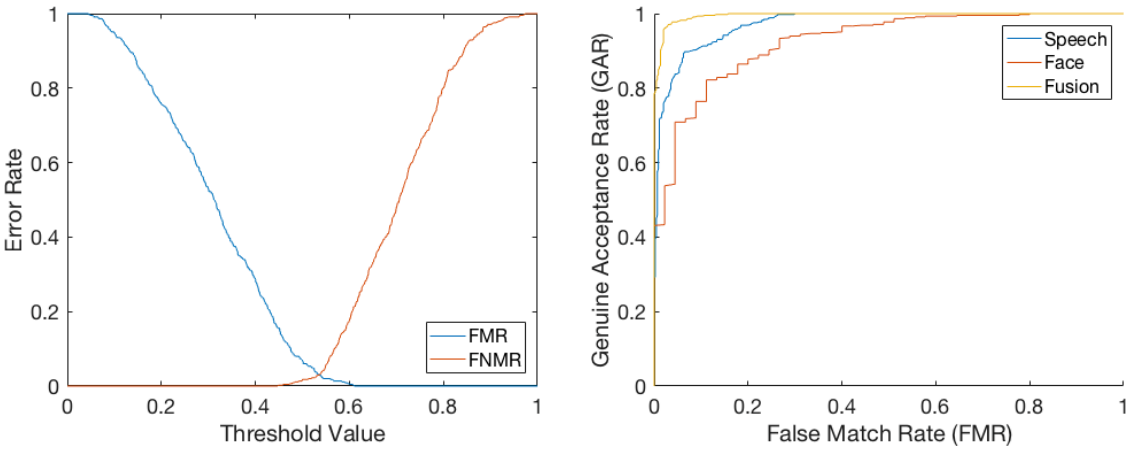


Figure 2.45 FMR and FNMR curves for the fusion method and ROC curves for the three systems (test set). Left: FMR and FNMR curves for the fusion method. Right: ROC curve for the fusion method, where the two modalities are equally weighted, compared with the ROC curves for the unimodal systems.

CHAPTER 3

ANALYTICAL PART

This part of the report contains a description of the company, the activities in which the student has participated, and some considerations regarding the whole experience.

3.1 Bang & Olufsen

Bang & Olufsen was founded in 1925 by Peter Bang and Svend Olufsen (Figure 3.1). Peter Bang was an engineer graduated at Aarhus Electrical Engineering, where he built several radio devices. Also Svend Olufsen was an engineer, and when he finished his studies he convinced Peter Bang to start a company in Struer, initially funded by Svend's mother, who raised sufficient funds from selling eggs.

Figure 3.1 Peter Bang (left) and Svend Olufsen (right).

The first commercial product launched by Bang & Olufsen was the Eliminator (Figure 3.2), a device that allowed the radios, which at that time worked with batteries that were expensive and impractical, to be connected to the mains. This was the start of the company success.

Figure 3.2 The eliminator.

Today Bang & Olufsen is well-known worldwide for the distinctive design and the innovative technology of its products. Bang & Olufsen is one of the most recognised brand in the world: it has been in the Top 20 CoolBrands list, made by The Centre For Brand Analysis to identify the UK's coolest brands, from 2008 until 2015. The current portfolio of products (Figure 3.3) includes:

- Wireless speaker systems.
- TVs.
- Speakers.
- Sound systems.
- Headphones.

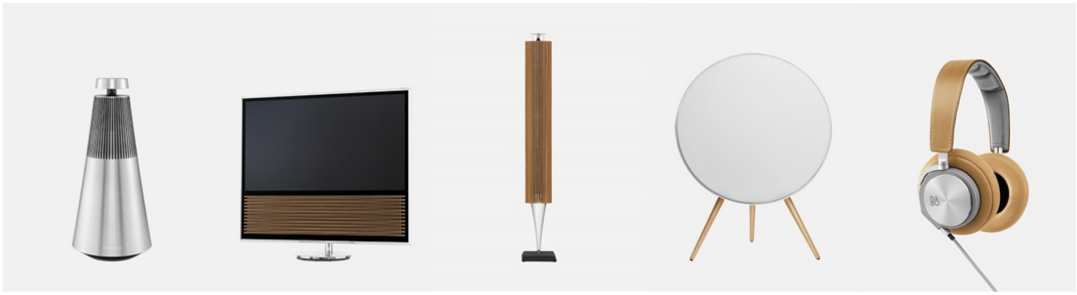


Figure 3.3 Some B&O products in the current portfolio. From left to right: BeoSound 2, BeoVision 14, BeoLab 18, BeoPlay A9, BeoPlay H6.

The company headquarters is in Struer, where also part of the development and the production takes place. The other production facility is located in Czech Republic, mainly focused on assembly and quality testing. The company site in Copenhagen deals with global sales and the B&O PLAY brand section.

Recently Bang & Olufsen has sold the automotive business to HARMAN, but it collaborates with Hewlett-Packard to bring the B&O sound to HP's products, and it started a partnership with LG Electronics on TVs and audio solutions for smartphones.

The organisational chart of the company is shown in Figure 3.4. It is possible to see that the COO area, whose role is to ensure efficiency and effectiveness of the business operations, is on top of the structure. Then we have five group functions (Portfolio & Program Management, Transformation, Group Operation, Group Quality & Service, and Group System Management), the Finance, and the Human Resource. The five business units where the company focuses are: Vision, Advanced Sound, Premium Sound, Smart Home, and Interaction. The three platforms that supports them are: Platform Development, Research, Maintenance; Cloud computing and Applications; Design, UX & Concept Exploration.

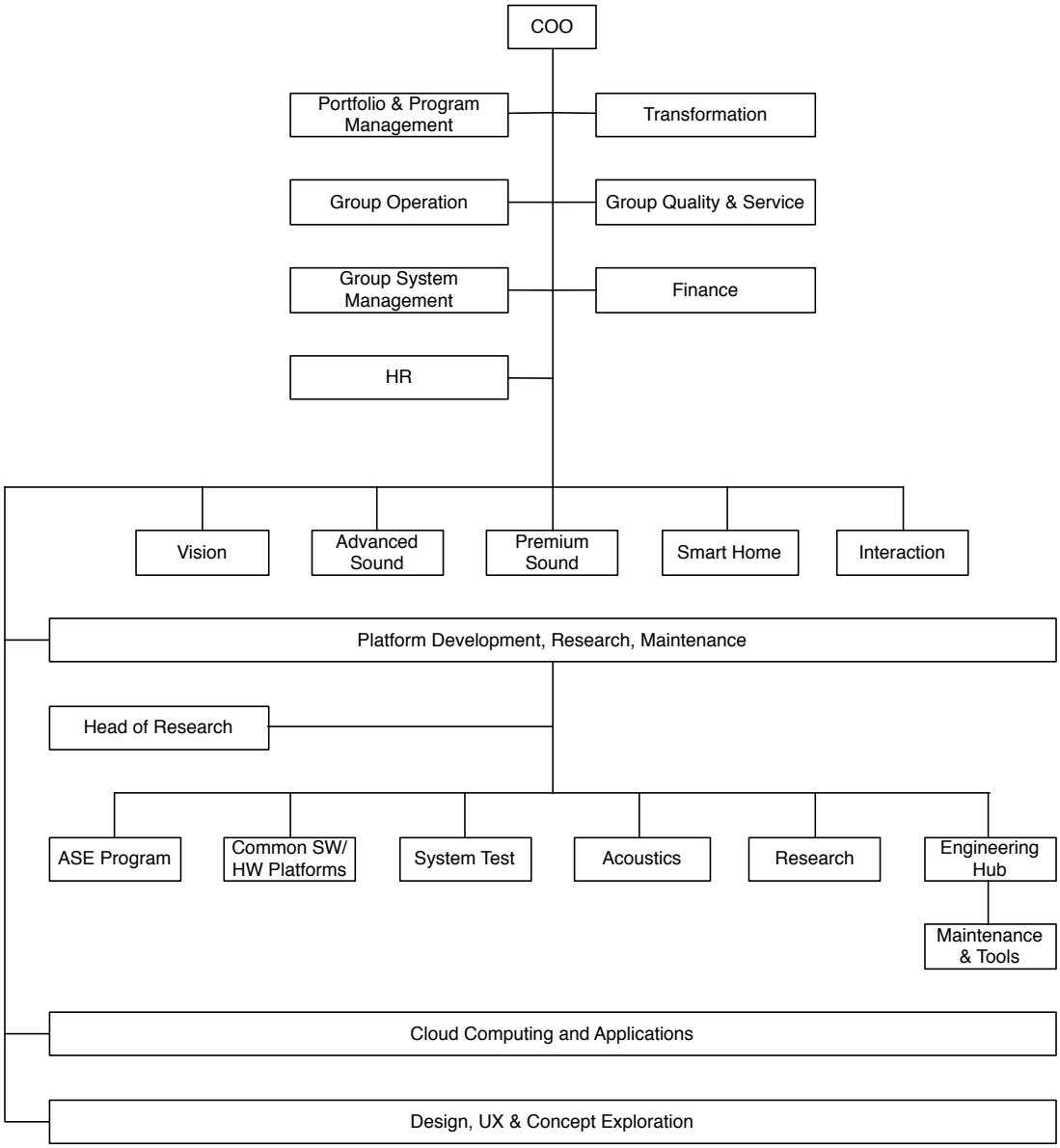


Figure 3.4 Organisational chart of the company.

The student worked in the research department, which is under the Platform Development, Research, Maintenance. The responsibilities of this department are:

- Establish, implement and maintain research strategy and competence roadmap to support product road map.
- Internal communication and alignment with key stakeholders.
- Write, coordinate and track funding applications.
- Initiate and support patent process.
- Establish and maintain collaboration with external partners (universities, companies).
- Represent B&O in relevant funding bodies and committees.

3.2 Tasks and Considerations

The main task of the student consisted of building an interactive prototype to offer a personalised user experience. He was not assigned to work on a side project, but to make an actual contribution. This highlights the level of participation, trust and involvement in the company. In order to improve the project, he had weekly meetings with his company supervisor where they discussed together about the main choices to make in each step. He always felt free about the main decisions, and this gave him the confidence to work independently. This was really important both academically and professionally, because it allows him to conduct research on the state-of-the-art approaches in the direction he wanted to pursue, and to use the tools he was interested in. The knowledge he gained during the study program was fundamental in this case, because it was the basis for the project.

The student also had the opportunity to work directly with the colleagues in the design, the UX, and the concept exploration departments, in order to have insights regarding the user experience that could be achieved with the prototype. He participated in a workshop with the purpose of deciding how the gaze-based system could be used to offer other kinds of personalised experience to the user. He presented the prototype during the *Ideas and Inspiration* session and the *End of Year* presentation to some Bang & Olufsen employees and was asked to take part in the filming of the demo for internal use.

He also participated in the weekly meetings of the research department. Here the current and future projects are presented and discussed, as well as the collaborations with universities and research institutes. During these meetings he could understand how the decisions are taken regarding different aspects and which factors have a greater impact.

He had a good experience also regarding the working environment and the relationship with his co-workers. The atmosphere at work was really informal and relaxed, and the social relations were ensured by various events that were regularly organised (such as having breakfast together once a week, participating in community meetings, gathering for a party etc.).

CHAPTER 4

CONCLUSION

In this report the performance of a multimodal identification system has been evaluated. We presented four techniques for face recognition (Eigenfaces, Fisherfaces, Local Binary Patterns Histograms, and Convolutional Neural Networks) implemented using OpenCV and OpenFace, and the i-vector approach for speaker recognition with SPro and ALIZE libraries. We performed the test on the AT&T face database, and on a multimodal dataset we built from several TED talks.

Our tests show that Convolutional Neural Networks outperform all the other feature extraction methods for face recognition, allowing to reach high performance. When the information from the camera is reduced, for example due to occlusion, the performance drops quite significantly, but good recognition performance can be achieved by using speech information.

The face recognition module of this system has been integrated in an interactive prototype, whose goal is to control a TV using gaze and offer a personalised user experience. Future works that can follow this study include:

- A complete implementation of the multimodal identification system for the prototype.
- A further study of the possibilities that this technology can open from a user experience point of view.
- The implementation of other modules (e.g. gestures) for an interaction that gives a more natural feeling.

ACKNOWLEDGEMENT

The author would like to thank both the university supervisor, Prof. Zheng-Hua Tan, and the company supervisor, Dr. Sven Ewan Shepstone, for the support during the whole experience.

He would also like to thank all the people in the research and the concept departments of Bang & Olufsen for having been great colleagues.

Finally, he would like to give special thanks to the other interns he spent good time with in Struer.

BIBLIOGRAPHY

[1] Lectures from the Toronto CSC class CSC2535: Computation in neural networks - variational learning. `http://www.cs.toronto.edu/~hinton/csc2535_03/lectures.html`, 2003.

[2] Notes from the Stanford CS class CS231n: Convolutional neural networks for visual recognition. `http://cs231n.github.io`, Accessed: 04-10-2016.

[3] TED: Ideas worth spreading. `https://www.ted.com`, Accessed: 04-10-2016.

[4] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pages 469–481. Springer, 2004.

[5] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[6] Tadas Baltru, Peter Robinson, Louis-Philippe Morency, et al. Openface: an open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2016.

[7] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 354–361, 2013.

[8] Woodrow Wilson Bledsoe. Man-machine facial recognition: Report on a large-scale experiment. Technical report PRI 22, Panoramic Research, Inc., 1966a.

[9] Woodrow Wilson Bledsoe and Helen Chan. A man-machine facial recognition system: Some preliminary results. Technical report PRI 19A, Panoramic Research, Inc., 1965.

[10] Jean-François Bonastre, Frédéric Wils, and Sylvain Meignier. Alize, a free toolkit for speaker recognition. In *ICASSP (1)*, pages 737–740, 2005.

[11] Gary Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] AT&T Laboratories Cambridge. The database of faces. `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`, Accessed: 04-10-2016.

[13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[14] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[15] OpenCV dev team. OpenCV 2.4.13.1 documentation: Histogram equalization. `http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html`, Accessed: 04-10-2016.

[16] OpenCV dev team. OpenCV 2.4.13.0 documentation: Support vector machines. `http://docs.opencv.org/2.4.13/modules/ml/doc/support_vector_machines.html`, Accessed: 04-10-2016.

[17] Hazim Kemal Ekenel, Mika Fischer, Qin Jin, and Rainer Stiefelhagen. Multi-modal person identification in a smart environment. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[18] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015.

[19] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[20] Guillaume Gravier. Spro: speech signal processing toolkit. *Software available at http://gforge.inria.fr/projects/spro*, 2003.

[21] Lars Hertel, Erhardt Barth, Thomas Kaster, and Thomas Martinetz. Deep convolutional neural networks as generic feature extractors. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–4. IEEE, 2015.

[22] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[23] Pavel Khlebovich. IP camera adapter 2.0. `http://ip-webcam.appspot.com`, Accessed: 04-10-2016.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Erfan Loweimi, Seyed Mohammad Ahadi, Thomas Drugman, and Samira Loveymi. On the importance of pre-emphasis and window shape in phase-based speech recognition. In *International Conference on Nonlinear Speech Processing*, pages 160–167. Springer, 2013.

[26] Davide Maltoni, Dario Maio, Anil Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2 edition, 2009.

[27] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015.

[28] Douglas Reynolds. Universal background models. *Encyclopedia of Biometrics*, pages 1547–1550, 2015.

[29] Toby Segaran. *Programming collective intelligence: building smart web 2.0 applications*. " O'Reilly Media, Inc.", 2007.

[30] Robert Snelick, Mike Indovina, James Yen, and Alan Mink. Multimodal biometrics: issues in design and testing. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 68–72. ACM, 2003.

[31] Samsung Support. How do i use picture in picture (pip) on my led tv? `http://www.samsung.com/hk_en/support/skp/faq/132767`, Accessed: 04-10-2016.

[32] Richard Szeliski. Image alignment and stitching: A tutorial. Technical report MSR-TR-2004-92, Microsoft Research, 2006.

[33] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[34] LG TV. Pip. `https://www.youtube.com/watch?v=DgQABsTxOTs`, Accessed: 04-10-2016.

[35] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[36] Erroll Wood, Tadas Baltruaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764. IEEE, 2015.

[37] Xiang Wu. Learning robust deep face representation. *arXiv preprint arXiv:1507.04844*, 2015.

APPENDIX A

WEEKLY JOURNAL

Week 1

- Presentation with the research group.
- Participation in the weekly meeting with the research group.
- Explanation by Sven regarding the internship project.
- Deadline for the final demonstration of the prototype (mid of December).
- Deadline for the first demonstration of the prototype (27th of September).
- Configuration of OpenCV 3.1.0 on a Windows 7 machine.
- Stream the video from the phone camera in OpenCV in order to work with a high resolution camera.

Week 2

- Participation in the weekly meeting with the research group.
- Participation in the monthly meeting with platform development, research, and maintenance.
- Environmental meeting.
- OpenCV face detection: use of Haar-cascade classifiers. Cons: false positive.
- OpenFace library: face detection using Dlib. Based on HOG+SVM: better detection. OpenFace also performs landmarks and gaze detection (+pose estimation).
- Getting started with face recognition using openCV. 4 possible ways: Eigenfaces; Fischerfaces; Local Binary Patterns Histograms; CNN (through dnn contrib library).

Week 3

- Participation in the weekly meeting with the research group.
- First implementation of face recognition based on eigenfaces.
- Documentation of the insights collected so far.
- Planning the collection of data for training/testing the system (face database).
- Participation in the presentation of an internship project about the C/MATLAB implementation of an online convolver.

Week 4

- Participation in the weekly meeting with the research group.
- First implementation of the face recognition module.
- First tests of the algorithms.
- Configuration of curl for possible use of online API.

Week 5

- Participation in the weekly meeting with the research group.
- First demo.

Week 6

- Participation in the weekly meeting with the research group.
- Study of the state-of-the-art of CNN for face recognition.
- Participation in the monthly meeting with platform development, research, and maintenance.
- Preparation of the presentation for the next research meeting.

Week 7

- Participation in the weekly meeting with the research group.
- Presentation of the project to the research group.
- Participation in the meeting about the revision of the research plan.
- First implementation of the User Interface.

Week 8

- Participation in the weekly meeting with the research group.
- Study of the face alignment problem.
- Implementation of a face alignment method based on three landmarks.
- Meeting with a UX designer in order to decide the main ideas for the final demonstrator from a UX perspective.
- Participation in the acoustics and research meeting.

Week 9

- Participation in the weekly meeting with the research group.
- Code refactoring of the face recognition module.
- Refined implementation of the User Interface.

Week 10

- Participation in the weekly meeting with the research group.
- Participation in the monthly meeting with platform development, research, and maintenance.
- Workshop on the UX possibilities of the project (with a demonstrator).
- First implementation of playback audio using two different devices (portaudio and libsndfile).

Week 11

- Participation in the weekly meeting with the research group.
- Refining the prototype for the demonstrator.
- Demonstration of the prototype in the ideas & inspiration session.
- Participation in the acoustics and research meeting.

Week 12

- Participation in the weekly meeting with the research group.
- B&O DNA Tour.
- Work on speaker recognition using Alizé.
- Christmas party.

Week 13

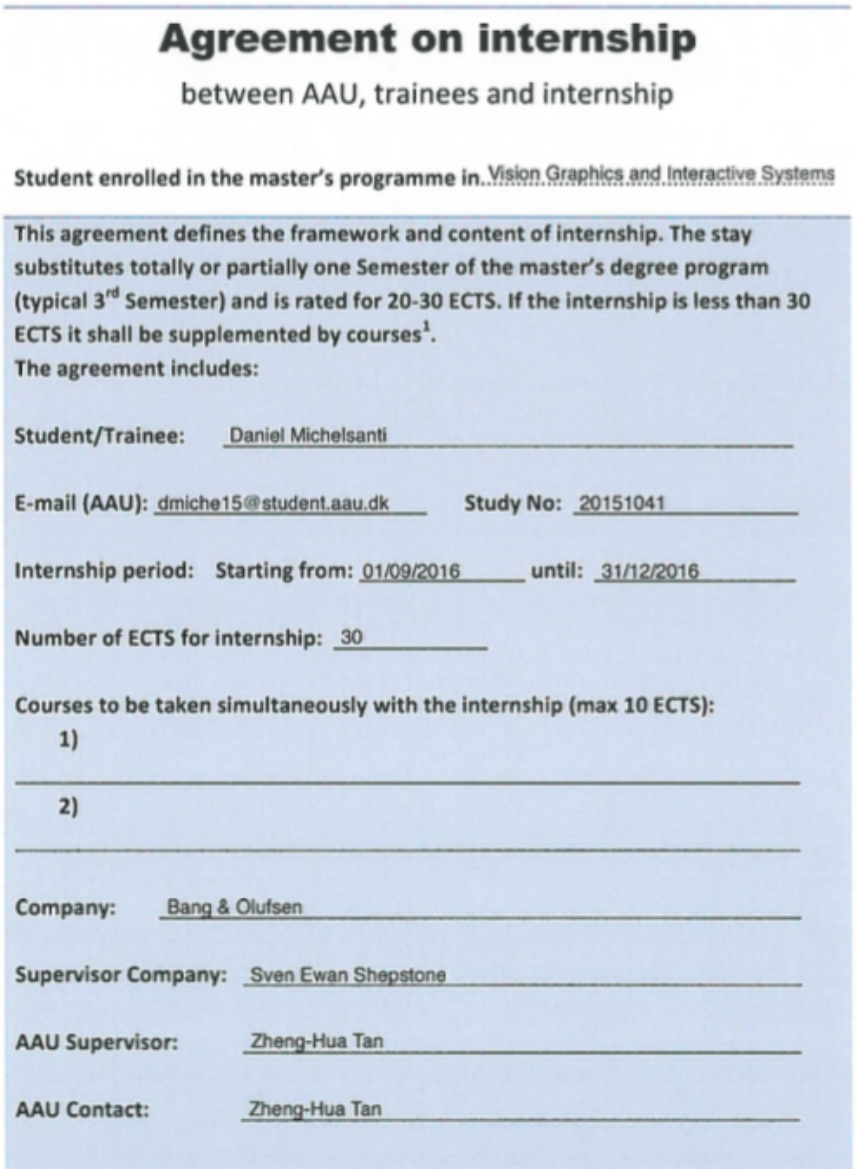
- Participation in the weekly meeting with the research group.
- Presentation of the project to the research group.
- Work on speaker recognition using Alizé.
- Filming of the demo.
- Participation in the acoustics and research meeting.

Week 14

- Participation in the weekly meeting with the research group.
- End of year presentation (demo).
- Work on speaker recognition using Alizé.

APPENDIX B

AGREEMENT ON INTERNSHIP



Agreement on internship

between AAU, trainees and internship

Student enrolled in the master's programme in. Vision Graphics and Interactive Systems

This agreement defines the framework and content of internship. The stay substitutes totally or partially one Semester of the master's degree program (typical 3rd Semester) and is rated for 20-30 ECTS. If the internship is less than 30 ECTS it shall be supplemented by courses[1].
The agreement includes:

Student/Trainee: Daniel Michelsanti

E-mail (AAU): dmiche15@student.aau.dk **Study No:** 20151041

Internship period: Starting from: 01/09/2016 **until:** 31/12/2016

Number of ECTS for internship: 30

Courses to be taken simultaneously with the internship (max 10 ECTS):

1)

2)

Company: Bang & Olufsen

Supervisor Company: Sven Ewan Shepstone

AAU Supervisor: Zheng-Hua Tan

AAU Contact: Zheng-Hua Tan

Documentation prepared by:

Project report ◯ *(tick)* **Assessment: Rated according to the 7-point scale**

Internship report ⊗ *(tick)* **Assessment: Pass / Fail**

Insurance:

The company has insurance for the student ⊗ **yes** ◯ **no** *(tick)*

(If the company has no insurance, it is up to the student to take out on)

The company's supervision during the stay [to be filled in by the company supervisor]

(Description of what the student can expect in terms of guidance and professional input. For example weekly meetings with the company supervisor and participation in review meetings)

Weekly meetings with Dr. Shepstone and Dr. Mortensen.
Close collaboration with other B&O collegues.

Assignments during the internship[to be filled in by the student in cooperation with the company supervisor]

(Description of what the student can expect in terms of involvement in different tasks).

Tasks in UX and SW department on designing sensor-based prototypes using a variety of techniques

Learning objective [to be filled out by the student in cooperation the AAU supervisor]

(As a starting point the goals of the curriculum is to be used; it can be adapted to the specific case. Learning objectives shall be divided into knowledge, skills and competencies]

Knowledge:

- must have knowledge about methods and architectures for fusion of information
- understand different aspects of user involvement in a particular system

Skills:

- analyze how a particular problem relates to an end-user
- suggest a solution that uses information derived from relevant modalities (vision, speech...)

Competences:

- communicate the above knowledge and skills (using proper terminology)
- select relevant theories, methods, and tools, and synthesize them in a new context

Internship report

During the internship the student keeps a diary, which forms the basis for the actual internship report, which will reflect the trainee's experience with the internship.

The internship report must include the following:

Description of the company, field of work and organization.

Description of tasks that the students have conducted and participated in during their stay.

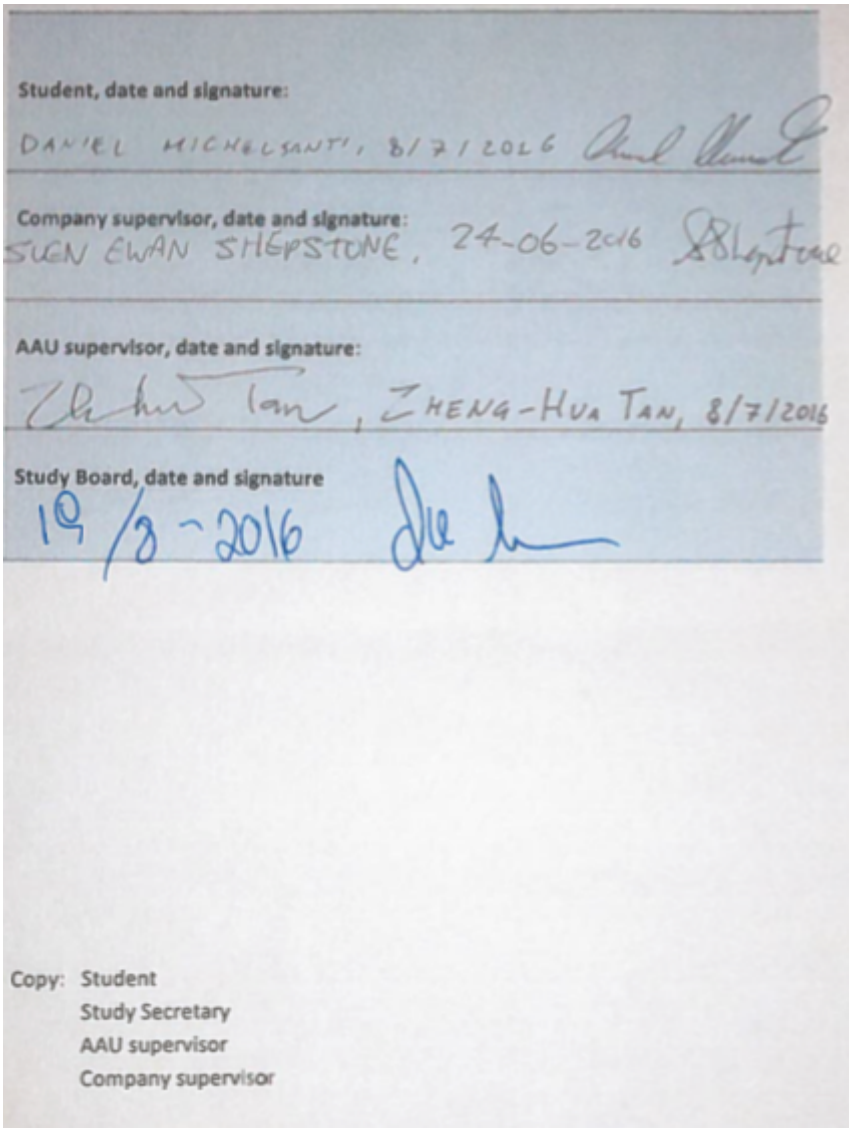
A particularly thorough description of one or two defined topics / areas of work that the students have dealt with during the placement. This will be in the form of the described tasks.

A conclusive evaluation of the overall internship, including reflection on the skills acquired, and the relation between applied theory and practice.

Written statement prepared by the place of internship site – (to be attached).

The statement prepared by the place of internship, and student diary will be foundation for the exam and the subsequent oral assessment, which is taking place immediately after the internship period, and is agreed between the supervisor and the student.

The internship report must be submitted within 14 days after the end of internship period.



Student, date and signature:

DANIEL MICHELSANTI, 8/7/2016

Company supervisor, date and signature:

SVEN EWAN SHEPSTONE, 24-06-2016

AAU supervisor, date and signature:

Zheng-Hua Tan, ZHENG-HUA TAN, 8/7/2016

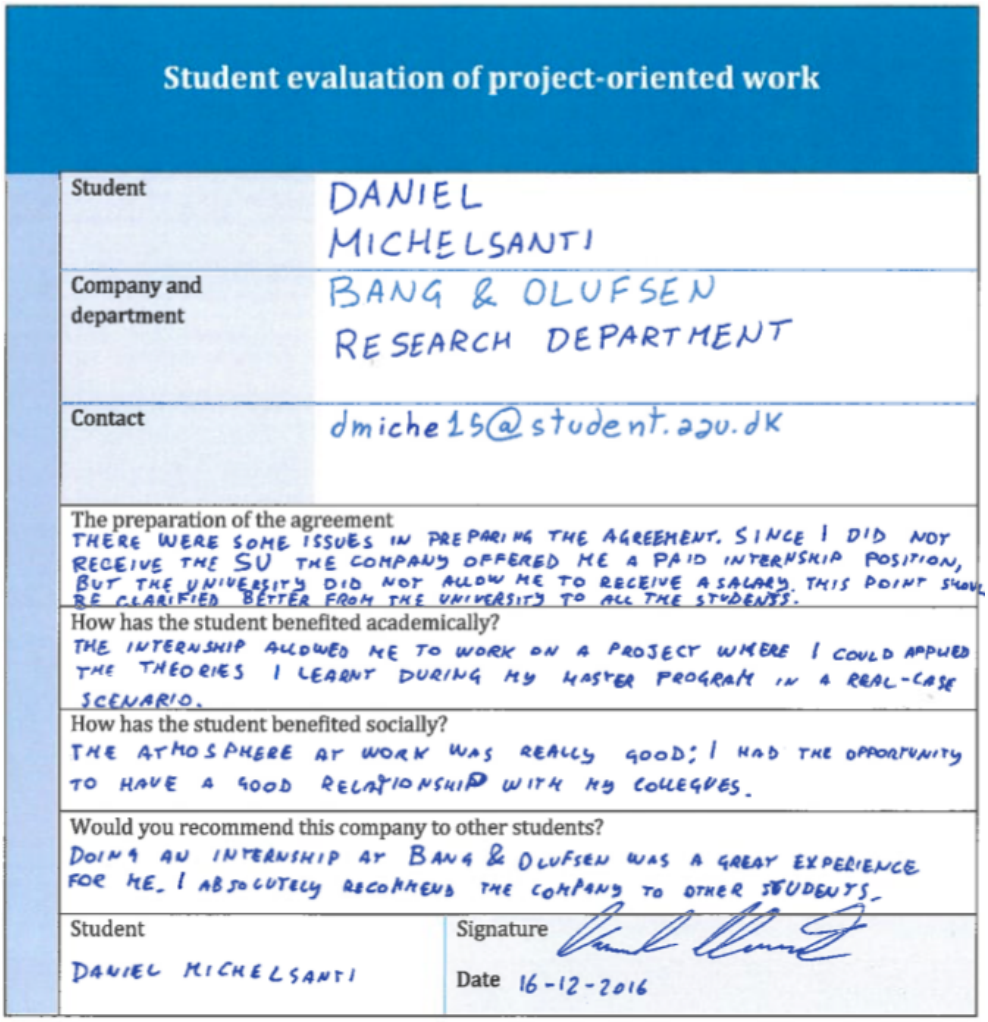
Study Board, date and signature

19/8-2016

Copy: Student
Study Secretary
AAU supervisor
Company supervisor

APPENDIX C

STUDENT EVALUATION OF PROJECT-ORIENTED WORK



Student evaluation of project-oriented work

Student	DANIEL MICHELSANTI
Company and department	BANG & OLUFSEN RESEARCH DEPARTMENT
Contact	dmiche15@student.aau.dk

The preparation of the agreement
THERE WERE SOME ISSUES IN PREPARING THE AGREEMENT. SINCE I DID NOT RECEIVE THE SU THE COMPANY OFFERED ME A PAID INTERNSHIP POSITION, BUT THE UNIVERSITY DID NOT ALLOW ME TO RECEIVE A SALARY. THIS POINT SHOULD BE CLARIFIED BETTER FROM THE UNIVERSITY TO ALL THE STUDENTS.

How has the student benefited academically?
THE INTERNSHIP ALLOWED ME TO WORK ON A PROJECT WHERE I COULD APPLIED THE THEORIES I LEARNT DURING MY MASTER PROGRAM IN A REAL-CASE SCENARIO.

How has the student benefited socially?
THE ATMOSPHERE AT WORK WAS REALLY GOOD; I HAD THE OPPORTUNITY TO HAVE A GOOD RELATIONSHIP WITH MY COLLEGUES.

Would you recommend this company to other students?
DOING AN INTERNSHIP AT BANG & OLUFSEN WAS A GREAT EXPERIENCE FOR ME. I ABSOLUTELY RECOMMEND THE COMPANY TO OTHER STUDENTS.

Student DANIEL MICHELSANTI	Signature Date 16-12-2016

APPENDIX D

COMPANY EVALUATION OF PROJECT-ORIENTED WORK

Company evaluation of project-oriented work

Company	BANG AND OLUFSEN A/S
Contact	SUEN EWAN SHEPSTONE
Student	DANIEL MICHELSANTI

The preparation of the agreement

IN THE BEGINNING THERE WAS SOME CONFUSION REGARDING THE TERMS (SALARY) OF THE STAY, BUT WE WERE ABLE TO SORT IT OUT

Cooperation with employees at Aalborg University

VERY SATISFACTORY

Assessment of the strengths and weaknesses of the study programme which the student has completed

STUDENT WAS VERY WELL-PREPARED TO TACKLE THE TASKS GIVEN. NO WEAKNESSES COME TO MIND.

The company's interests and wishes regarding future cooperation with the students and employees of Aalborg University

THE COMPANY WISHES TO CONTINUE WITH COOPERATION.

How might we optimise project-oriented work in the future?

NO RECOMMENDATIONS FOR IMPROVEMENT AT THIS STAGE

Company Contact	Signature
SUEN EWAN SHEPSTONE	Date 12-12-2016

Bang & Olufsen a/s | Peter Bangs Vej 15 | DK-7600 Struer | Denmark
Phone: +45 96 84 11 22 | CVR-nr.: 41257911
Bank: Nordea. Account 2149 8976 595 441
Iban DK70 2000 8976 5954 41

Struer, 2. januar 2017

To whom it may concern,

Daniel Michelsanti was employed on a full-time attachment at Bang & Olufsen from the 1st September 2016 to the 31st December 2016, both dates inclusive. The purpose of the project related work at Bang & Olufsen was to fulfil the requirements of the degree of Master of Science in Vision Graphics and Interaction Systems at Aalborg University, for which Daniel is currently enrolled.

While at the company, Daniel was tasked with developing machine learning technology to determine the identity of people for personalisation purposes. During the time of his stay, he researched and implemented four successful face identification algorithms, including a state-of-the-art deep learning network algorithm. In addition to this, he worked on speaker recognition. Daniel made a substantial contribution in developing a UX prototype for show-casing future interaction possibilities for television frameworks.

It was a pleasure to supervise Daniel. He is a keen learner, works independently, and is committed to the task in hand. At an important demonstration event, where a number of key strategic stakeholders were invited, Daniel successfully and confidently demonstrated his work, as well as how it could be applied to a future B&O product. In his daily activities at the company, he got on well with the staff and was well-liked.

I would gladly recommend Daniel to a future employer. Please do not hesitate to contact me if there are any questions.

Yours faithfully

Sven Ewan Shepstone (Company Supervisor)

BANG & OLUFSEN