



**Universidade do Minho**  
Escola de Engenharia

Pedro Nelson Sampaio Barbosa

## **Automotive Facial Driver Recognition**

Master's Thesis

Master in Industrial Electronic and Computers Engineering

Work developed under supervision of

**Professor Jorge Cabral**

January 2017



# Author Declaration

**Name:** Pedro Nelson Sampaio Barbosa

**Electronic address (email):** a65366@uminho.pt

**Phone:**+351 914601267

**Citizen Card:** 14363151

**Dissertation Title:** Automotive Facial Driver Recognition

**Advisor:** Professor Jorge Miguel Nunes Santos Cabral

**Conclusion Year:** 2017

Master in Industrial Electronic and Computers Engineering

AUTHORIZED INTEGRAL COPY OF THIS DISSETATION ONLY FOR PURPOSES OF RESEARCH, BY DECLARATION OF INTERESTED WRITTEN, THAT SUCH COMMITS.

AUTHORIZED INTEGRAL COPY OF THIS DISSETATION (indicate, if necessary, maximum number of pages, illustrations, graphics, etc.), ONLY FOR PURPOSES OF RESEARCH, BY DECLARATION OF INTERESTED WRITTEN, THAT SUCH COMMITS.

ACCORDING TO EXISTING LEGISLATION, IT IS NOT ALLOWED ANY PART REPRODUCTION OF THIS THESIS/WORK.

Minho's university, \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

Signature:

*“The man of knowledge must be able not only to love his enemies but also to hate his friends.”*

Friedrich Nietzsche

Friedrich Wilhelm Nietzsche (1844–1900) is a German philosopher of the late 19th century who challenged the foundations of Christianity and traditional morality



# *Agradecimentos*

A realização desta dissertação não teria sido alcançada se não fosse o apoio, compreensão e motivação de pessoas muito importantes e essenciais.

Primeiramente, gostaria de agradecer todo o apoio dado pelo meu orientador, Professor Doutor Jorge Cabral, que procurou estar sempre disponível para responder a qualquer necessidade que eu tivesse, bem como oferecer a sua opinião sincera sobre todo o trabalho realizado.

De seguida gostaria de agradecer a todos os meus colegas e amigos de laboratório, Artur Faria, Kevin Costa, Diogo Cruz, Rui Paixão, António Ribeiro e Pedro Leite, por toda a descontração, motivação e divertimento, que permitiram gerar um clima fantástico de trabalho. Não me poderia esquecer de dar um especial agradecimento ao meu colega/amigo Bruno Mendes pelo esforço extra que se dispôs a realizar, para a revisão e análise do documento redigido durante esta dissertação, a ele o meu "Muito Obrigado!".

Este trabalho foi financiado pelo projeto "INNOVCAR: Inovação para Veículos Inteligentes", n02797, cofinanciado pelo FEDER através do Portugal 2020 – Programa Operacional Competitividade e Internacionalização (COMPETE2020), e gostaria de agradecer a oportunidade que me foi dada de nele participar. Gostaria de dar um especial agradecimento a dois membros deste projeto que ofereceram uma grande dedicação na supervisão deste trabalho, Marco Martins e Margarida Monteiro. Um obrigado por toda a ajuda dada e pelo apoio nos momentos em que a motivação faltou.

Apesar de não terem contribuído diretamente para a realização desta dissertação, gostaria de agradecer ao meu círculo de amigos que me ouviram, apoiaram, motivaram e até forneceram dicas para uma melhor escrita do meu inglês, o meu obrigado a Bruno Costa, Francisco Fernandes, José Palha e Bruno Ferreira.

Finalmente quero agradecer às pessoas que me são mais importantes e que definiram o que sou hoje, e sem elas não era capaz de chegar a este momento. Aos meus adorados pais, Maria de Lurdes e António Barbosa, por toda a motivação e compreensão que me deram durante todo o desenvolvimento deste trabalho, e pela possibilidade dada para a ingressão neste curso, e à minha irmã, Rita Barbosa, a mulher de quem tenho mais orgulho e me deu maior apoio nos momentos em que surgiram obstáculos pessoais durante a realização deste projeto.

Um muito obrigado a todos!



MINHO UNIVERSITY

# *Abstract*

Engineering School  
Department of Industrial Electronic

Master in Industrial Electronic and Computers Engineering

## **Automotive Facial Driver Recognition**

by Pedro Nelson Sampaio Barbosa

Dependency on technology is quite real when it comes to the automotive world. Currently there are several automated systems that can be found in cars: lights control, seats control, brakes control and the list goes on. These systems enhance the safety of drivers but raise other issues that need to be addressed.

In the automotive market, when trying to stand out, some brands seek to attract consumers through the originality and provide options that were never seen before in their vehicles. Nowadays some vehicles have the option to change some functionalities depending on the driver's will, such as seat and steering wheel height, suspension/motor mode, among others. But with this level of changes a question arises: will the driver have to make these changes every time he uses the car? The answer is no, because if the user somehow performs a driver's identification, the car system can load the personal customizations made by that particular person.

This work realize the driver's recognition automatically and with the minimum of hindrance. For this purpose it is acquire an image of the driver's face in order to know his identification. This system will recognize the driver and send his identification to an external system, so that this second system can perform the required customization. The system to build in this dissertation have to be small in size, quickly realize the driver's recognition and to inform her identity to an external system. The necessary process for the use of this system must be simple, and allow the graphic visualization of the whole process that is taking place.

The final system aims to increase the technical knowledge of the "Project INNOV-CAR", which is a project resulting from the partnership between the Minho's University and the renowned multinational in the automotive world, Bosch. All the development and conclusions taken during this work will revert to this project, and the system to be developed also has the possibility of being integrated into the DSM (Driver Simulator Mockup). The DSM is a simulator of an automobile in a virtual world, in which all the systems that compose it, use the TCP/IP protocol suit as a means of communication. The final system to be developed, in this dissertation, should use the TCP/IP protocol suit, as a means of receiving commands and sending the driver's identity.

**Keywords:** Driver Recognition, Customization Load, Bosch Car Multimedia, INNOV-CAR, Minho's University



UNIVERSIDADE DO MINHO

# *Resumo*

Escola de Engenharia  
Departamento de Eletrónica Industrial

Mestrado em Engenharia Eletrónica Industrial e Computadores

## **Automotive Facial Driver Recognition**

por Pedro Nelson Sampaio Barbosa

A dependência pela tecnologia é algo muito real quando se trata do mundo automóvel. Atualmente podem ser encontrados muitos sistemas automáticos num carro: controlo das luzes, controlo dos assentos, controlo dos travões e a lista continua. Estes sistemas aprimoram a segurança dos condutores nos seus veículos, mas levantam outros problemas que têm de ser resolvidos.

No mercado automóvel, de forma a se destacarem dos restantes, algumas marcas procuram atrair o consumidor através da inserção de opções e originalidade, nos seus veículos, nunca vistas anteriormente. Atualmente alguns veículos têm opções para alterarem algumas funcionalidades, dependendo do gosto do condutor, opções como altura dos assentos ou volante, estações no menu de acesso rápido, entre outras. Mas com o aumento da quantidade de alterações que se pode realizar num veículo, uma questão surge: o condutor terá sempre que realizar essas alterações sempre que vá usar o carro? A resposta a essa questão é não, pois se de algum modo for realizada uma identificação da identidade do condutor, o sistema do carro pode realizar o *load* das personalizações associadas a um específico condutor.

Este trabalho pretende realizar o reconhecimento da identidade do condutor de forma automática, com o mínimo de desconforto possível. Para isso é proposto a aquisição de uma imagem da face do condutor, de forma a realizar a sua identificação através dela. Este sistema irá realizar o reconhecimento do condutor, e de seguida enviar a sua identidade para um sistema externo, para que este segundo sistema possa adquirir as personalizações anteriormente feitas, pelo condutor em questão. O sistema a desenvolver nesta dissertação propõe ser de pequenas dimensões, realizar o reconhecimento rapidamente e enviar a identidade obtida, para um sistema externo. O processo necessário para usar este sistema deve ser simples, e permitir a visualização gráfica de tudo o que está a acontecer.

O sistema final tem como objetivo o aumento do conhecimento crítico do "Project INNOVCAR", que se trata de um projeto resultante da parceria entre a Universidade do Minho e a empresa de renome mundial no mundo automóvel, Bosch. Todo o desenvolvimento e conclusões executadas/retiradas durante este trabalho revertem para este projeto, e o sistema a desenvolver tem também a possibilidade de ser integrado no DSM (*Driver Simulator Mockup*). O DSM é um simulador automóvel num mundo virtual, e todos os sistemas que o compõem, usam o protocolo TCP/IP como forma de comunicação. O sistema final a desenvolver, nesta dissertação, também deve usar o protocolo de comunicação TCP/IP, como meio de receção de comando e envio da identidade do condutor.

**Palavras-chave:** Reconhecimento do Driver, *Load* das personalizações, Bosch Car Multimedia, INNOVCAR, Universidade do Minho



# Content

<b>Author Declaration</b>	<b>iii</b>
<b>Agradecimientos</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Resumo</b>	<b>ix</b>
<b>Content</b>	<b>xi</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>Abbreviations List</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contextualization . . . . .	1
1.2 Motivation and Goals . . . . .	2
1.3 Academic Contribution . . . . .	2
1.4 Research Questions . . . . .	3
1.5 Requirements . . . . .	3
1.5.1 Image Acquisition . . . . .	3
1.5.2 Data Processing . . . . .	4
1.5.3 Data Comparison . . . . .	4
1.5.4 Server Communication . . . . .	4
1.5.5 Main System . . . . .	4
1.6 Risks . . . . .	4
1.7 Dissertation Outline . . . . .	5
<b>2 State of the Art</b>	<b>7</b>
2.1 Driver Identification . . . . .	7
2.1.1 Types of identification . . . . .	8
2.1.2 Manual Identification . . . . .	9
2.1.3 Auto Identification . . . . .	9
2.1.4 Identification Card . . . . .	10
2.1.5 Biometric Identification . . . . .	11
2.2 Case study . . . . .	15
2.2.1 Visible Spectrum . . . . .	16
2.2.2 Infrared Image . . . . .	16
2.2.3 3D Image . . . . .	18
2.2.4 Type of Image . . . . .	21
2.2.5 Face Recognition Techniques . . . . .	22

2.3	State of the Market . . . . .	28
2.3.1	Volvo System . . . . .	29
2.3.2	BMW . . . . .	30
2.3.3	Ford and Intel . . . . .	31
2.3.4	Chinese Police . . . . .	32
2.3.5	Harman . . . . .	33
2.3.6	Peugeot and Citroen . . . . .	33
<b>3</b>	<b>Methodology and System Specification</b>	<b>35</b>
3.1	Methodology . . . . .	35
3.2	System Requirements . . . . .	37
3.2.1	Functional Requirements . . . . .	37
3.2.2	Non-functional Requirements . . . . .	37
3.3	System Overview . . . . .	37
3.3.1	Loading System Overview . . . . .	38
3.3.2	Facial Recognition System . . . . .	39
3.4	Technology Specification . . . . .	39
3.4.1	Tecnhnology Criteria . . . . .	40
3.4.2	Technology Definition . . . . .	40
3.5	Hardware Specification . . . . .	41
3.5.1	3D Cameras . . . . .	41
3.5.2	Camera Definition . . . . .	43
3.5.3	CPU Handling . . . . .	44
3.6	Software Specification . . . . .	44
3.6.1	Programming Languages Selection . . . . .	45
3.6.2	Software Development Platforms . . . . .	45
3.6.3	Development Platforms Selection . . . . .	45
3.6.4	Algorithms Libraries . . . . .	46
3.7	System Architecture . . . . .	47
3.7.1	Local System . . . . .	47
3.7.2	Server System . . . . .	54
3.7.3	GUI Application . . . . .	58
3.8	Validation Specification . . . . .	62
3.8.1	Validation Criteria . . . . .	62
3.8.2	Tests Specification . . . . .	63
3.8.3	Test Method . . . . .	63
<b>4</b>	<b>System Implementation</b>	<b>65</b>
4.1	Final Prototype System . . . . .	65
4.1.1	Tests Configuration . . . . .	65
4.1.2	Materials for the Prototype System . . . . .	65
4.2	System Architecture . . . . .	66
4.2.1	Local System . . . . .	67
4.2.2	Server System . . . . .	81
4.2.3	GUI Application . . . . .	84



<b>5</b>	<b>Results and Validation</b>	<b>89</b>
5.1	Initial Tests . . . . .	89
5.1.1	OpenCV . . . . .	90
5.1.2	Intellisense . . . . .	91
5.2	GUI Application . . . . .	91
5.2.1	GUI Components . . . . .	91
5.2.2	GUI Connections Warnings . . . . .	92
5.2.3	GUI/Local System Normal Cycle . . . . .	94
5.2.4	GUI/local System Multi Face . . . . .	98
5.3	Server System . . . . .	99
5.3.1	Multiclient Handling . . . . .	99
5.3.2	Communication Protocol . . . . .	99
5.4	Local System . . . . .	101
5.4.1	Users Database . . . . .	102
5.4.2	Measurements . . . . .	102
5.4.3	Search Interval . . . . .	103
5.4.4	Compatible Percentage . . . . .	103
5.4.5	Update Measurements . . . . .	103
5.5	Database Content . . . . .	104
<b>6</b>	<b>Conclusions and Future Work</b>	<b>105</b>
6.1	Conclusions . . . . .	105
6.2	Future Work . . . . .	106



# List of Figures

2.1	Global car sharing members [1]	8
2.2	BMW Profiles [2]	8
2.3	RFID Tag [3]	10
2.4	Example Biometric measures	11
2.5	Example of a ECG result[4]	12
2.6	Ford's Heart Beat Seat(2011) [5]	13
2.7	Human Fingerprint Points [6]	13
2.8	First Driver Facial Recognition System idea using a camera [7]	15
2.9	Image acquire in Near-infrared wave [8]	17
2.10	Image acquired in Long-infrared wave [9]	17
2.11	Same point acquired from two different cameras [10]	18
2.12	Configuration types to make Laser Triangulation [11]	19
2.13	System diagram for Projected Light 3D Image [12]	19
2.14	System diagram for TOF 3D Image [13]	20
2.15	OCT exam from a eye [14]	21
2.16	Video Sequence Methods	22
2.17	First set of images, example [15]	23
2.18	Second set of images, example [15]	23
2.19	Facial parts separation	24
2.20	Head rotation record [16]	24
2.21	Accuracy when the number of points or number of frames change [16]	25
2.22	Intensity images methods	25
2.23	Eigenfaces Example [17]	26
2.24	Brunelli and Poggio 35 face measurements [18]	27
2.25	Gaile Gordo approach using 3D images [19]	28
2.26	Brands/Companies distribution	29
2.27	Volvo Face Recognition System [20]	30
2.28	BMW face recognition system [21]	31
2.29	Ford and Intel Face Recognition System [22]	32
2.30	Chinese Face Recognition System [23]	33
2.31	Peugeot face recognition system [24]	34
3.1	Adopted Methodology	36
3.2	System Overview	38
3.3	System Overview	38
3.4	Complete Facial Recognition System	39
3.5	Camera IntelSense SR300 [25]	42
3.6	Camera Kinect [26]	42
3.7	Camera mvBlueSIRIUS [27]	43
3.8	Mini PC-Intel NUC Kit [28]	44
3.9	System Design	47
3.10	Main Local System Flowchart	48

3.11	Closest User Identification Flowchart . . . . .	49
3.12	Face Calibration Flowchart . . . . .	50
3.13	Face Measurements Flowchart . . . . .	51
3.14	Face Checking Flowchart . . . . .	52
3.15	Recognition of a new Driver Flowchart . . . . .	53
3.16	Start Recognition Sequence Diagram . . . . .	54
3.17	Good Name Sequence Diagram . . . . .	55
3.18	Error Name Sequence Diagram . . . . .	56
3.19	Server System Threads . . . . .	56
3.20	Server System Main Thread Flowchart . . . . .	57
3.21	Server System Thread Client Flowchart . . . . .	57
3.22	Graphical GUI Mockup . . . . .	58
3.23	GUI System Architecture . . . . .	59
3.24	Image Thread Flowchart . . . . .	59
3.25	Start Identification Button Thread Flowchart . . . . .	60
3.26	Regist Button Thread Flowchart . . . . .	61
4.1	Laboratory Setup . . . . .	66
4.2	Final System Diagram . . . . .	66
4.3	Camera vision axes [29] . . . . .	73
4.4	Head pose [29] . . . . .	75
4.5	The 20 face measures acquired . . . . .	77
5.1	OpenCV results to the challenge . . . . .	90
5.2	Intellisense results to the challenge . . . . .	91
5.3	GUI components analyze . . . . .	92
5.4	Camera not connected (Warning) . . . . .	93
5.5	TCP connection not made (Warning) . . . . .	93
5.6	Face position indications . . . . .	94
5.7	Face recognition available . . . . .	95
5.8	Face recognition of a new user . . . . .	95
5.9	Writing the user's name . . . . .	96
5.10	User registration . . . . .	97
5.11	Face recognition available again . . . . .	97
5.12	Face recognition of known user . . . . .	98
5.13	Multi face detection . . . . .	98
5.14	Server multiclient handling . . . . .	99
5.15	Protocol to known user . . . . .	100
5.16	Protocol to unknown user . . . . .	101
5.17	Database contest after the tests . . . . .	104

# List of Tables

2.1	Types of image comparison . . . . .	21
3.1	Type Images Comparison . . . . .	40
3.2	Comparison between cameras . . . . .	43
4.1	Stream Types to SR300 Intel camera . . . . .	69
4.2	TCP connection progress regist . . . . .	71
4.3	Arrays Content before Measure1 check . . . . .	79
4.4	Arrays Content before Measure2 check . . . . .	79
4.5	TCP Command sended from local system . . . . .	84
4.6	TCP Command sended from server . . . . .	84
5.1	GUI components . . . . .	92
5.2	System accuracy based on users database . . . . .	102
5.3	System accuracy based on measurements calculated . . . . .	102
5.4	System accuracy based on measurements interval . . . . .	103
5.5	System accuracy based on identical measurements . . . . .	103
5.6	Update measurements after recognition . . . . .	103



# Abbreviations List

<b>HMI</b>	<b>H</b> uman <b>M</b> achine <b>I</b> nterface
<b>TCP</b>	<b>T</b> ransmission <b>C</b> ontrol <b>P</b> rotocol
<b>IP</b>	<b>I</b> nternet <b>P</b> rotocol
<b>RFID</b>	<b>R</b> adio <b>F</b> requency <b>I</b> dentification
<b>ID</b>	<b>I</b> Dentification
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem
<b>ECG</b>	<b>E</b> lectro <b>C</b> ardio <b>G</b> ram
<b>CES</b>	<b>C</b> onsumer <b>E</b> lectronics <b>S</b> how
<b>3D</b>	<b>3</b> <b>D</b> imensional
<b>IR</b>	<b>I</b> nfra <b>R</b> ed
<b>TOF</b>	<b>T</b> ime <b>O</b> f <b>F</b> light
<b>OCT</b>	<b>O</b> ptical <b>C</b> oherence <b>T</b> omography
<b>AI</b>	<b>A</b> rtificial <b>I</b> ntelligence
<b>LED</b>	<b>L</b> ight <b>E</b> mitting <b>D</b> iode
<b>DSM</b>	<b>D</b> river <b>S</b> imulator <b>M</b> ockup
<b>CPU</b>	<b>C</b> entral <b>P</b> rocessing <b>U</b> nit
<b>SSD</b>	<b>S</b> olid <b>S</b> tate <b>D</b> rive
<b>PC</b>	<b>P</b> ersonal <b>C</b> omputer
<b>USB</b>	<b>U</b> niversal <b>S</b> erial <b>B</b> us
<b>RAM</b>	<b>R</b> andom <b>A</b> ccess <b>M</b> emory
<b>GUI</b>	<b>G</b> raphical <b>U</b> ser <b>I</b> nterface
<b>OS</b>	<b>O</b> perating <b>S</b> ystem
<b>FPS</b>	<b>F</b> rames <b>P</b> er <b>S</b> econd
<b>DB</b>	<b>D</b> ata <b>B</b> ase
<b>SQL</b>	<b>S</b> tructured <b>Q</b> uery <b>L</b> anguage
<b>HMM</b>	<b>H</b> idden <b>M</b> arkov <b>M</b> odel





# Chapter 1

## Introduction

In this first chapter, it is possible to read the contextualization of the project, the motivations and objectives, as well as the main conclusions. It will be presented the contribution to the academic world. Since this project focuses on a recent technology, it will also address the risks that can compromise the non-resolution of this project and the necessary requirements for its conclusion. Finally, a structure of the thesis will be presented in this chapter, with a small explanation on the focus of each chapter.

### 1.1 Contextualization

The first project involving the recognition of human faces using computers was in the years 1964 and 1965. The pioneers were Charles Bisson, Woody Bledsoe and Helen Chan Wolf. Financed by an unknown intelligence agency, they built a semi-automatic system able to calculate unique measurements in every person and store this data in a database for future comparisons. The only disadvantage of this system was the need of the presence of an administrator (real person) to identify the mouth, nose, eyes and ears in the acquired photographs, which only allowed the analysis of 40 photos per hour [30].

Face recognition technology had great importance during the Super Bowl of January 2001. Technology developed by Matthew Turk and Alex Pentland allowed through video surveillance images, the identification of 19 people with arrest warrants, by comparing the acquired images with a database of wanted people.

The use of this technology is not limited to identification of wanted people, but can also be used for more common needs, such as entry into specific spaces or restricted access locations, presence register in a specific place or possible identification and customization of a setup in an automobile, which consists in a major goal of this dissertation.

It is public knowledge, an automobile is not restricted to a single driver, i.e., multiple drivers can use the same car, and each one may have different preferences. One might like to have the seat and steering wheel a little bit higher, while others prefer the music volume a bit higher than normal and other drivers may like the dashboard with a different color.

With the increase of possible car interior customization, or even outside of a car, the need to somehow speed up the process and save customization made by a driver is essential. To do that, it is necessary to store the customizations made by the driver, for when the driver decides to pick up the car again, these customizations are loaded and applied, thereby preventing the driver from always do his preferences. One of the quickest ways is to keep in a custom system (inside the car) the customization associated

with a user, thus the next time the driver enter the car, he would only need to select (manually) the corresponding user profile.

Facial recognition can help the driver identification issue, avoiding the creation of a user profile to store data associated with the driver, regarding several features such as the level of the seat, steering wheel, volume, type of music. This way the driver needs only to enter the car and drive, because the moment the driver enters the vehicle, the "car" will know who is driving it and adjust itself to the driver's preferences.

## 1.2 Motivation and Goals

In the modern era, the dependence on technology is large. One example is the daily use of a mobile phone or a laptop, and this dependence greatly affects the automotive sector, which has to keep up with trends in order to attract new customers.

This fact requires the evolution of automotive industry, leading to investments in development and production of advanced multimedia solutions and, with that, the emergence of new technologies.

These new technologies emerge with the main objective to improve the driving experience of drivers, so that the distraction period of the driver for this action is as small as possible, increasing their safety. However, sometimes this goal is not reached, so it is required a different approach to solving this problem.

Increasing the possible customizations inside, or outside, of a car, is not only expecting differentiated designs and set-ups, but also increasing comfort and taste for the driving feeling of the driver. After some implementations and studies, it was conclude that the driver searches for customization in the hour of buying a vehicle.

This dissertation will focus on the implementation of a system that facilitates and speeds up the storing of the customizations applied by a driver, thus loading these settings the next time the driver enters the vehicle. This system cannot affect the existing vehicle design in a significant way and must perform the identification of the driver in a short period of time. This dissertation will be realized in partnership with a company that is well-established in the automotive world, Bosch Car Multimedia. The final results expected with the resolution of this problem, such as percentage of correct answers in driver's identification, will be compared with standard values already defined by the company, to check if the final implementation is feasible in the automotive world. The solution presented is a camera that will capture an image of the driver's face and compare it with drivers who have been in the car to see if it is a new user or not. At the end it is expected that the system correctly identifies the driver to therefore load yours customization more easily.

The partnership will not only allow comparison with the standards in the automotive world, but also acquire essential skills for the business world, namely the ability to communicate with other professional areas, adopt the language and behavior appropriate to each situation and the consolidation of knowledge already acquired and improve skills of English language speaking.

## 1.3 Academic Contribution

From the partnership between the University of Minho and the Bosch company in 2013, the project "INNOVCAR: The Cockpit of the Future" was born, and all the developments and results obtained in this dissertation will contribute to this project. This dissertation will contribute with the resolution of these questions:

- Design and implementation of new HMI interfaces, as well as new ways to identify the driver to implement a system of self-customization, that is, customization of the setup of a vehicle automatically without the need for the user to enter data directly;
- Exploration of new technologies for possible implementation in the automotive world;
- Identification of the driver's gaze to know, his or her, greatest area of attention for placing warning signs about the road or the vehicle.

## 1.4 Research Questions

The research questions are the basis of this project and all the procedures, research and studies have been made in order to answer them. Just below these lines, they are written in the chronological order in which they will appear in this report. As it has already been said, they will be answered during this thesis.

- First question: What is the importance of driver identification?
- Second question: How can it be done?
- Third question: Using only face recognition is enough?

## 1.5 Requirements

To increase the efficiency, construction and understanding of the final developed system, it was divided into four parts: image acquisition, data processing, data comparison and communication with the server. For the development some requirements are defined, some only for completion of one of the phases, but others have to be done for completion of the entire project.

### 1.5.1 Image Acquisition

- The system must collect a driver image;
- Has to collect the image with minimal noise;
- Does not have to depend of physical characteristics of the driver.

### 1.5.2 Data Processing

- The system must be accurate enough to identify a face on an image;
- Process all data in short time(less then 1 second);
- The hardware necessary to processing the data, must occupy a small space.

### 1.5.3 Data Comparison

- The system must load the information of a database;
- Compare the image acquired with a database.

### 1.5.4 Server Communication

- The system has to connect with a server using TCP/IP protocol suit;
- Must answer to commands received from server;
- Send the driver identification to server.

### 1.5.5 Main System

- The system must be compact;
- Be independent of any external system;
- Recognize the driver in short time;
- Work with an offline database.

## 1.6 Risks

For the correct recognition of the driver, it is necessary that the acquired image has a high quality in order to reduce the processing time necessary for the recognition of the face. Since in this thesis the system is intended for a possible installation in a vehicle, it has to identify the driver in darkness, as well as in the presence of light. The system has, however, some risks, such as:

- Because it is necessary a camera that acquires images to allow the identification of the driver both in the presence of visible light and in its absence, the impossibility of finding a camera with small dimensions capable of doing that has a high weight in the conclusion of the system;
- Because it is necessary to analyze images with great quality for the correct identification of the driver, a high power processing may be necessary, which can restrain the space needed for the system and its independence from an external system;
- The identification of the driver must be done in a short period of time, but due to the need for a large data acquisition, this factor can be a restriction;

- A poor communication between the system and the server can restrain a correctly operation of the system. For that, it is necessary a reliable and quick communication in order to reduce delays and prevent loss of information.

## 1.7 Dissertation Outline

This thesis is organized in six chapters, namely:

- Introduction (Chapter 1)

As in all dissertations, the first chapter aims to give a little introduction to what it is expected of the project: the reason why it has been made, what objectives it has, what issues are expected to be resolved, the requirements, and the risks it presents. It also presents the structure of the dissertation;

- State of the Art (Chapter 2)

This chapter will present the state of the art of the project to be developed. Initially a theoretical approach will be made to the technologies used in this project, as well as other technologies also used for the same purpose (identification of the driver in an automatic way). After the theoretical approach, real cases of implementation will be presented using the technology studied in this project. Images will be presented as well a brief summary of the functioning of these systems that can already be found in market, or are still under development;

- Methodology and System Specification (Chapter 3)

The presentation of the methodology used in the project as well the specification of all the necessary systems for the completion of the project will be made in this chapter, chapter three. Here it will be defined the system architecture, the necessary software and hardware, the final conception design, and how the system will be validated. Also here it will be explained some choices relative to software and hardware used;

- System Implementation (Chapter 4)

This chapter will present the implementations chosen in the previous chapter, as well a brief explanation of how they were achieved. Here it is also presented some excerpts of code with explanations of why and what they do in the main code, as well the messages between TCP server and system, used to communicate with each other;

- Results (Chapter 5)

In chapter five i are presented the results of the implementations used to reach the final presentation, as well as the results of the final prototype. Some comparisons will be made between them and it is also explained the improvements implemented until reaching the final stage;

- Conclusions/Future work (Chapter 6)

In this last chapter, it will be presented some conclusions about the work developed in this dissertation, as well as the results achieved. An analysis on some possible future approaches that would lead to an improvement of the final system is also shown.



# Chapter 2

## State of the Art

This chapter will present some theoretical elements of the main theme of this dissertation, such as the importance of the main objective for the automotive world, what kind of inputs and algorithms can be used to implement it, as well as systems in development, or already developed, capable of accomplishing the objective proposed in this dissertation. Firstly, other technologies that can achieve the same objective (that identification of the driver) will be presented. After a discussion about the type of images, and algorithms, that can be used to identify the driver. Finally, the current state-of-the-art in research and similar commercial solution will be presented.

### 2.1 Driver Identification

The automotive industry is one of the largest and most important in the world. Nowadays it is almost impossible to go through a place and not see at least one car, since the need and utility of a car in our lives has been increasing. A study carried in 2016, shows that car production has been growing every year, an approximate growth of 66% in 2016 compared to the year 2000 [31].

This growth not only leads the industry itself to produce even more to respond to the consumption demand, but also leads each brand to find a way to stand out from the rest, by introducing new technologies and options in their vehicles [32].

Interior customization of a vehicle is something that has already become normal in some car brands, but as the number of possible modifications increases, a problem arises with this increase, such as, how to avoid the driver having to make the customization every time he uses the car. Some brands already tried to solve this problem by giving different keys for each user of the vehicle, like in model i8 from the brand BMW. Others placed small switches next to the driver's seat to switch between different profiles, so that when the user made changes such as height of the headlights, bank height, change color of the central stack and others, he would not have to make those changes again.

By inserting these solutions into their vehicles, another problem arose. As shown in the figure chart 2.1, the number of members using car-sharing applications has been increasing a lot since 2012, and it is expected to continue increasing for the following years. And with the increase of car sharing, the solutions previously identified to "recognizing" the driver's preferences are not feasible, because if the vehicle is shared by four or five drivers, the need for multiple keys would increase, thus increasing the probability of key loss, as well as possible car theft. This led to the creation of new systems capable of storing a large number of profiles without requiring dependencies from external hardware, and also to increase security.

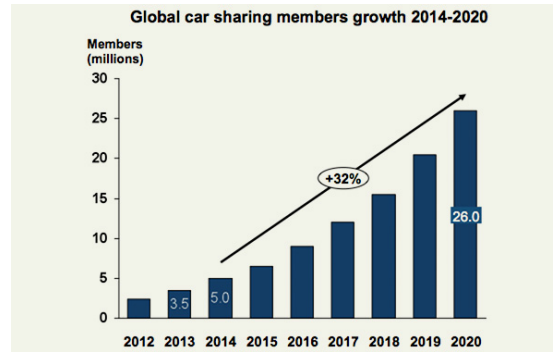


FIGURE 2.1: Global car sharing members [1]

These new systems are able to save several profiles associated with a car's interior customizations, basically keeping in an internal memory a profile associated with that customization and enabling the user to re-access them if he/she wishes. These systems seek to access these settings in a manual way, where the driver selects which profile he/she would like to load, or automatically, where a system in the car identifies the driver and load the associated profile.

The use of these systems allows for the creation of multiple profiles, limiting their number only to the available memory inside the car.

### 2.1.1 Types of identification

As mentioned in the section "Driver Identification", there are two ways to identify the driver in order to load the associated profile in the car. The load can happen when the user has selected the desired profile in one of the menus of the car, as seen in the models marketed this year, 2016, in the BMW brand (as it is possible to see in the figure 2.2), or through an automatic mode, which will be referred in examples below.



FIGURE 2.2: BMW Profiles [2]

The manual identification has its advantages and disadvantages, with one of the advantages being the possible creation of several profiles by the same user. Using the radio as an example, since one of the possible customizations on the car is the fast access to radio stations, so when the driver is in good mood, he can select stations with more upbeat and energetic music, and save them in one profile, but if he is in bad mood, he can select stations with relaxed atmosphere musics, and save them in another profile.



This way the driver has two profiles for different moods, one for when he is in a good mood, and another for when he wants to relax.

As for disadvantages the manual identification does have one, namely when the driver has to navigate between menus to select the desired profile he wants to load. Even so, these systems can be much faster and practical than the older systems, since the driver had to choose the settings every time he used the car.

Regarding systems that make the automatic profile loading associated to a driver, they lack the problem of only allowing the association of one profile for each user, but they have the advantage that they load the profile in a very fast way without needing any navigation through menus. How this is done will be described in following sections.

## 2.1.2 Manual Identification

The appearance of the first car with profile selection associated with customization, was in 1981 by Cadillac. In this car it was possible to choose between two profiles previously defined by the driver. The only custom changes saved were just the ones made to the driver's seat. It was only in the 1990's that they released models with the same system that was able to save changes in rearview mirrors, brake and accelerator pedals, climate control and the car stereo system [33]. The profiles were selected on a small switch located on the left side of the driver's seat, typically alongside the seat adjustment controls.

Since then many car models have been launched with identical systems, capable of saving much more settings than described previously and allowing a greater selection of profiles, but the principle of operation remains the same. Its operation has pleased consumers so much that vehicles are launched with identical systems [34], with the difference that the switch has been replaced with a menu in the central stack of the vehicle, since many of the cars launched today come equipped with control displays to change their options.

## 2.1.3 Auto Identification

The automatically profile loading on the car system without requiring direct input from the driver is a fairly recent technology. Only in 2008 did a system appear that can perform the identification of the user without having to make any selection in one of the menus of the car, or press any option on a switch. The main purpose of this system was not the loading of previously settings made but the identification for safety reasons.

At this moment, systems can be found/seen on the market that identify the driver through identification cards or through biometric measurements of the driver.

In the first case, identification card, the principle of operation is identical to that previously described in section "Driver Identification", when the BMW i8 model example is mentioned. These cards are unique to each driver and have the advantage of being much cheaper than a car key, compensating a big production of this cards if the vehicle is used by multiple drivers.

In the case of identification by biometric measurements, the driver does not have to perform any task. The entire process involved in the identification is carried out without the driver having to provide any device. Not many systems can be found, and even in

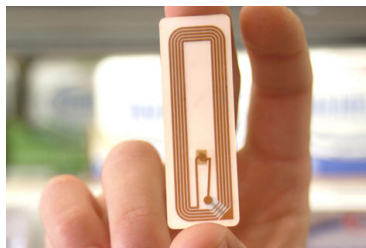
those that are, some are still under development. The case under study in this thesis falls into this category.

### 2.1.4 Identification Card

A very simple method to identifying a person is for that person to possess some kind of tag or ID card and present it to the system he/she wants to access. Through this label the system obtains information about its owner and verify if the owner has access. This method is already commonly used in many places, like in schools for student access to specific areas, hospitals for permission of restricted access places or even in car parks for permission to enter with the car.

A Portuguese company, Frotcom, started in 2008 the sale of a service that aims to control a fleet of vehicles using RFID technology, or Radio Frequency Identification, as the basis for the driver identification [35]. After the system installation in the vehicle to read RFID cards, the driver who intends to use the vehicle must present his RFID card in the reader and if he has authorization, the vehicle will start the ignition. It is a very simple method that not only reduces the possibility of theft, since it is very difficult to cheat the system if the driver does not have an authorized card, but also allows the owners of the fleet of vehicles to verify and analyze who used each vehicle and the route that it made. Incorrect identification of the driver or march started without proper identification, leads to the activation of the vehicle alarm, alerting the people around of a possible theft.

RFID is a method of automatic identification through radio signals. The information we want to store is kept safe on devices called tags (figure 2.3). Typically they have the appearance of a card and the data is acquired by RFID readers.



---

FIGURE 2.3: RFID Tag [3]

An tag is a transponder, which is a small silicon chip and antenna that respond to radio signals sent by a transmitter base. The base sends a radio signal that causes the response of the RFID tag, where this response can have information such as driver name and access level.

## 2.1.5 Biometric Identification

One of the methods that can be used to identify a person is to compare their biometric data with that stored in a system. Biometric data are measurements related to the physical characteristics of the human body. This data is unique and can be used in systems of identification and access control [36]. Some examples of biometric measurements can be seen in the figure 2.4.

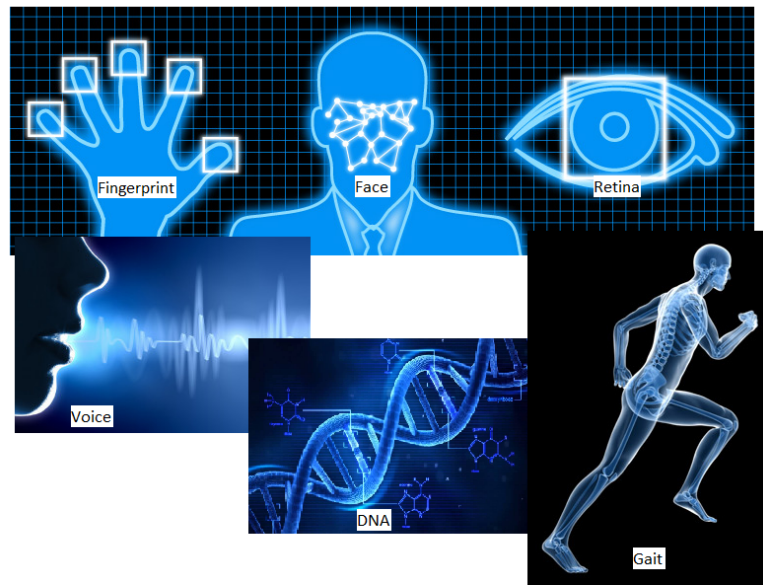


FIGURE 2.4: Example Biometric measures

The biometric data is distinct and unique for each individual, so it is possible to trace a profile of each person with them, thus allowing access to locations or sensitive information without resorting to the use of passwords or any other means of identification. This data can be divided into two categories: physical biometric data and behavioral biometric data. Physical biometric data is related to the physical characteristics of a person. These characteristics are practically impossible to find in another individual, such as DNA, fingerprint, hand pattern, face geometry, iris geometry, among others [37]. Behavioral Biometric data is related to the behavior of a person, something that was developed during his life, characteristics such as writing, voice and way of walking. The focus in this dissertation will be on the physical biometric data, since the recognition of an individual by its face, falls into this category.

Many other methods can be used for the identification of a specific individual by a system such as the use of RFID, or others such as using unique documents like a citizen's card. But all these methods have a problem, they all depend on the memory of their users. In the case of ID cards, the user has to remember to always pick them up, and the use of passwords involves that the user always remembers the password when he wants to use it. In the case of biometric data, the user does not have to do anything, since he walks with his "identifier" every day, he only has to appear in the place where his biometric data will be necessary. The use of this data can not only be used in safety matters but also in health issues, since this data can identify an injured driver in the car, and send an alert to nearby hospitals in the area [38].

Some projects involving the identification of the driver by their biometrics measurements are already developed, some are in the development stage, while others have reached the final prototype. During this dissertation it will only be approached biometric measurements where there is already existing development made for the automotive area.

## Heart Beat

A unique characteristic in every human being is our heart beat. It is a signal acquired by an "exam" called the ECG, electrocardiogram, and may seem at first glance that the resulting signal is similar in some cases to the other patients, but it is not true. By analysing characteristics such as amplitude, shape and frequency of the signal, it is possible to see that each person has a different and unique heart beat. This singularity makes possible the use of this signal as a means of identifying a person, since its acquisition does not involve a very long procedure, and it is not invasive [4].

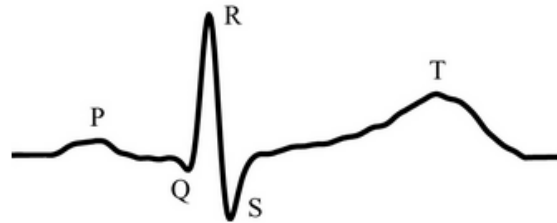


FIGURE 2.5: Example of a ECG result[4]

In the figure 2.5 it is possible to see the result of an ECG, and in it the three phases of a cardiac cycle are identified. The P wave represents the atrial depolarization and contraction, the QRS complex represents the depolarization and contraction of the ventricles and finally the T wave represents the repolarization of the atria. In other words, the P wave represents the opening of the valves in the heart for the blood to entry, the QRS complex, represents the closure of the valves and contraction of the ventricles, and the T wave the release of blood and relaxation of the ventricles [39]. The association of an ECG signal to an individual is possible, since the whole wave remains unchanged even if the individual is in great physical effort, changing only the frequency of the same and the time of the appearance of the T wave, but the amplitudes of the waves, format and time duration, remains unchanged [4], allowing a secure identification.

Like all technologies, identification by heart beat has some disadvantages. The largest of them is the acquisition of the signal. In order to avoid the use of non-invasive means to perform the ECG, moistened electrodes are placed in the chamber to increase the conductivity of the place where they are placed. Obviously the use of these electrodes does not cause discomfort, but it is not a very practical solution in the automotive field. Another problem is the difficulty in acquiring the signal when the subject is pregnant, or has some metal object.



---

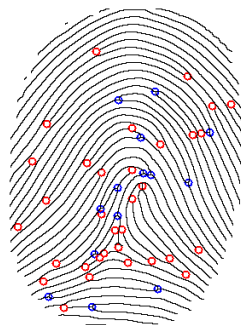
FIGURE 2.6: Ford's Heart Beat Seat(2011) [5]

In 2011 Ford announced the development of a system for the automotive industry, capable of identifying the driver through his heart beat (figure 2.6). This system would be able to perform the ECG without direct contact with the driver's skin, thus reducing the previously described discomfort. Unfortunately in 2015 the company announced the cancellation of the project due to the existence of more accurate and cheap systems in the market, such as wearables from Apple, Fitbit and Jawbone [5].

Another company developed a project based on this biometric measurement to identify the driver, so that the car would know who the driver was. The company CardioID was born in 2014, a year after having had the idea of putting electrodes on the steering wheel in order to perform an ECG and that way identify the driver. This idea came to be implemented and tested by the project "INNOVCAR: The Cockpit of the Future" in that same year.

## Fingerprint

The impression left by the friction between a human finger and a surface, fingerprint, is an excellent method for identifying a driver. This discovery made in 1903 [40] on surfaces and inside gloves, has been used to identify possible suspects or victims of a crime by forensic scientists.



---

FIGURE 2.7: Human Fingerprint Points [6]

Fingerprints are formed by the transfer of substances from the human finger to a surface when come into contact, resulting in a unique pattern perfectly capable of identifying a person (figure 2.7). If another fingerprint is found, then it can be compared

to check who it belongs to. These points may be a complete finger print or, more typically, only a portion. They are a feature that hardly changes throughout life, making this biometric measurement ideal for long-term identification.

In the market it is possible to already find a device of easy installation, that stops the car ignition from people whose fingerprint is not registered and authorized in the system [41].

Other several technologies have already been developed in order to acquire the human fingerprint, all of them using non-invasive and fast methods. They can be captured by using a digital image, by ultrasound, or by using capacitive sensors:

- Optical: Using a specialized digital camera, a digital photo is taken from the fingerprint of the person in question. A light is emitted on the finger. The light reflected by the finger is transmitted to pixels that captures the image. This procedure has some disadvantages. Because it is an image, it is subject to the possible visibility of the finger, that is, if for some reason the finger is dirty, or the skin is damaged, the image quality will be degraded, not allowing the correct acquisition of a fingerprint;
- Ultrasonic: By emitting high frequency waves on the finger or hand, it is possible to recreate a virtual image through the echo generated by the waves in the hand. It is not a very expensive process and is fast running. This process has advantages compared to the previous one, since the waves are only reflected in the dermic layer of the skin, the layer responsible for joining the skin to the subcutaneous, that way eliminating the need of a clean skin of the finger or palm, and also eliminating errors caused by damages on the skin itself;
- Capacitance: The fingerprint can still be detected through capacitive sensors. By measuring the capacitance of the person's fingerprint, it is possible to recreate an unique array of values for each person. Capacitance is a quantity determined by the amount of electrical energy that can be accumulated in itself, and as each person is different, when measuring the capacitance of a finger, through an array of sensors, it is possible to associate this array of values to each person and use it as an identification method.

## Voice

One of the main means of communication used between people is speech, and from that it is possible to recognize someone else without even seeing them. Features such as trim voice, accent or even expressions used during a speech can help one person identify another. Because of these characteristics, the interest in using the voice to identify a person is increasing.

Nuance, founded in 1992 in the United States, specializes in the development of image processing and speech recognition software. One of their projects aimed at speech recognition is the Nuance Dragon Drive, which can be found today in the new BMW 7 Series models. This allows the interaction between vehicle and driver in a quick and easy way, using this to acquire and analyse the driver's speech. A phrase said by the driver as "Drive me to the street Azurem Campus" activates the GPS that indicates the way forward to go to the street Campus of Azurem. This technology came to be present at CES 2016, which is an event for the presentation or demonstration of new technologies

for buyers and investors to acquire, which occurs every year, and in this event, Nuance system awarded the prize of CES 2016 Innovation Awards Honoree.

In speech recognition systems, two types of approaches can be used, "speaker independent" or "speaker dependent". Speaker dependent is when the system is trained to recognize certain words or phrases, using as a sample the phrase spoken several times by a single person, identifying the text spoken by a specific person, or just the text when spoken by several people. Speaker independent is when there is no training, with the speech being segmented using a small list of known words, thus not allowing an identification with a more accuracy compared to the first approach, but a greater freedom in the discourse [42][43].

For the system development capable of identifying a driver through his voice, inside a car, the better approach would be the "speaker dependent", with training to recognize only the specific driver. Such system will always have a problem in introducing new authorized drivers, because it would involve a new training, which can be expensive due to the processing power necessary to train the network.

## 2.2 Case study

As already stated in previous sections, the identification of the driver can be made through biometric measurements of the driver. Some examples have already been given in the section 2.1.5 and many others could be used, but the one that was chosen for this dissertation, is the use of a camera to acquire an image of the driver's face for later identification and recognition.

Another identification could be made by acquiring an image, and recognizing the driver by their iris. To do that, it is necessary to focus the image more on the eyes and the photo needs to have a great resolution to make the identification possible. The basic principle of an iris identification is the recognition of the unique color pattern around the pupil of the eye, joining the single reflection to the wave spectrum that each eye has, due to the different concentrations of protaines. Every person has a diferent color pattern and diferent reflection response to a wave spectrum [44].

The identification of the driver using a camera is a recent idea. The first patent registered on the subject appeared in 2009, but goes back to 1996 [7], and notes the idea of installing a camera on the back of the car to identify the driver by picking up his reflection in a mirror in front of the vehicle (figure 2.8).

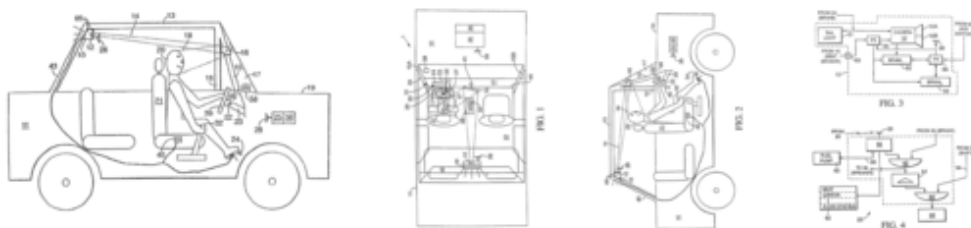


FIGURE 2.8: First Driver Facial Recognition System idea using a camera [7]

In the registered patent it is said that the identification will be made using cameras that capture images in the infrared wave, and the identification would be used to substitute the key necessary for the ignition of the car. The system would, if a valid driver was identified, start the ignition of the car, not involving the need for any key.

As noted in the patent, images of the driver's face are used to identify him, but images in visible light may also be used, or depth images, i.e., 3D images. The use of each of them depends very much on the conditions that the system will be to carry out the identification. Issues such as luminousness, actuation time or accuracy will affect what type of image is needed and which approach/algorithm to implement.

The following sections will present the types of images that can be obtained, the advantages and disadvantages of each image type, how they can be obtained and what types of algorithms, used to identify and recognize a person, can be found for each type of image.

### 2.2.1 Visible Spectrum

Of the three types of images that can be used for driver identification, the simplest is the a image in the visible spectrum. It is an image collected in the spectrum of visible light to the human eye and is acquired by using an image sensor. Consisting of an array of photodiodes, it converts light into a set of electric charges that are then translated into a digital image [45].

Photodiodes are semiconductor devices that convert light into an electric charge [46]. If placed in matrix format, it is possible to acquire the image, and the number of photo diodes in the same area, defining the resolution of the acquired image. If the number of photo diodes is high, the resolutions of the imagem is high to. The unit assigned to the resolution of the image is dpi (dots per inch), the screen that displays it is ppi (pixel per inch) and the unit of the sensor that acquired the image is samples per inch [47].

### 2.2.2 Infrared Image

Infrared images are those that are obtained at light wavelengths 700 nm to one millimeter, and in this wavelength two types of infrared images can be obtained. The mode of acquisition of the infrared image is the same as the one to acquire a visible light image, but with the difference where an infrared filter is placed, that is, a filter that allows the infrared light to pass to the camera, but blocks all or the greater part of the visible light spectrum [9].

#### Near-infrared

It is a region of great interest in electronics because it is in this zone that remote controls, object sensors and distance detectors operate. Being a zone that works a little above visible light, it is very easy to operate at this wavelength. The Near-infrared band extends up to 1400nm. The common emitter wavelengths are 850nm and 950nm. Due to its easy use in this range, there is many interference around us that can affect the acquisition of signals or images in this zone, such as cameras, lights or controls (figure 2.9).





FIGURE 2.9: Image acquire in Near-infrared wave [8]

### Long-infrared

It is light at wavelength 8000nm-15000nm. It is a thermal imaging area, and it is possible to acquire a thermal color image, where vivid colors represent high temperatures and the absence, low temperature color (figure 2.10).

This range of light is much appreciated in astronomy for the identification of gaseous formations or planets in the universe, with stars being captured in full vivid colors thanks to their immense temperature. The absence of color in the middle of this image may represent a rocky formation or a gaseous formation unknown until the moment [48].



FIGURE 2.10: Image acquired in Long-infrared wave [9]

### 2.2.3 3D Image

The most complicated and difficult image to acquire is the 3D image, since for each coordinate point image we have the 3 axis (X, Y, Z) and its acquisition usually involves a great processing power. For the acquisition of a 3D image, there are five types of technology that can be used to acquire the image: stereo vision, laser triangulation, projected light, flight time sensors and optical coherence tomography [49].

#### Stereo Vision

Stereo vision is the technology that comes closest to the way someone identifies a 3-Dimensional object. Two cameras (emulating eyes) are installed separately with a distance between them, observing the same object but with different perspectives (figure 2.11). Due to these two perspectives, it is possible to calculate the distance and re-create the object digitally.

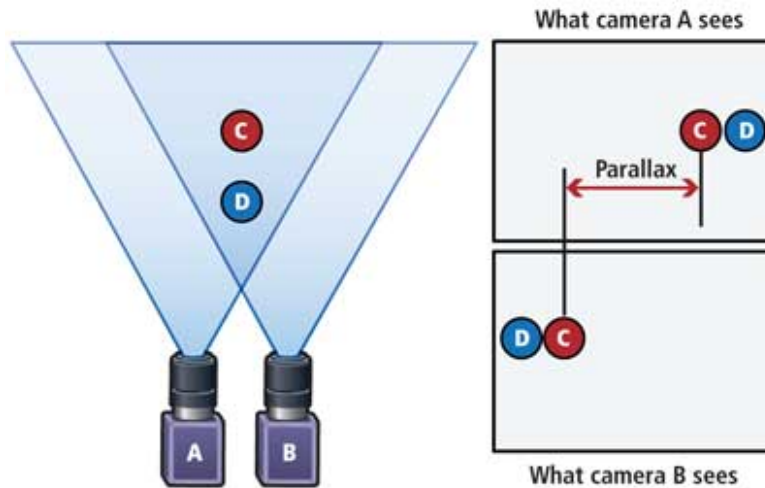


FIGURE 2.11: Same point acquired from two different cameras [10]

The principle behind this technology is that points of the same object will be represented differently in each of the cameras. A point in the left camera will suffer a pixel deviation in the right camera. By subtracting this difference it is possible to recreate a difference map [49].

With this map plus the distance that each camera is from the object and angle view, it is possible to recreate a 3D model of the object. Because this process involves only capturing an image in each of the cameras to create the 3D model, it is ideal for situations where objects are moving quickly.

#### Laser Triangulation

3D modeling technology through laser triangulation is easy to understand, but depending on the assembly type, can be difficult to implement (figure 2.12).

To create a 3D image using this technology it is necessary a laser, a camera and for these two components to always remain with the same distance and angle towards the target object throughout the process. Basically a strip of narrow light is focused on an object, and the camera films the deformation that the strip has in the object.

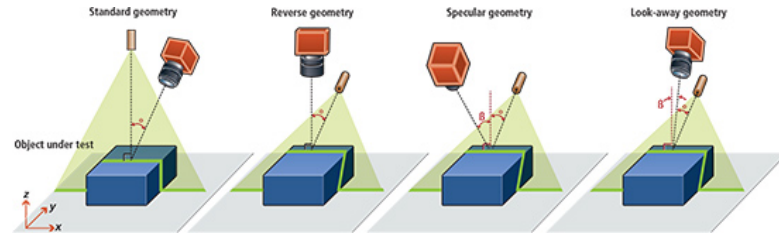


FIGURE 2.12: Configuration types to make Laser Triangulation [11]

Each type of assembly has advantages and disadvantages, and for this reason each one of them is used for different objects. To perform a 3D modeling with high resolution and precision, the second type of assembly is used, reverse geometry, but it has the disadvantage of requiring high processing power. For modeling reflector objects, such as glass and metal, the fourth type of assembly is used, and for dark objects or great texture, the third type is used. The first type of assembly (standart geometry) is used for generalized cases that require non-high resolutions and previsions [11].

## Projected Light

This type of 3D modeling technology is almost an improvement on the previously mentioned technology. In the target object a known pattern of light is projected, and the deformation of the light pattern in the object is recorded on an camera positioned on an known angle and distance, this way the displacement in each pixel, caused by the object, in the original light pattern is calculated. With this information plus angle and distance from the camera to the object, it is possible to create the 3D image of the same. In the figure 2.13, it is possible to see the system overview of a scan that uses this technology.

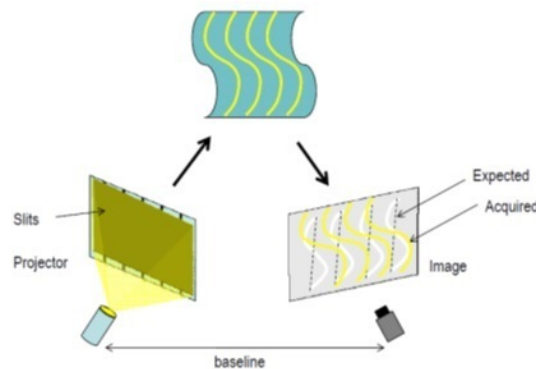


FIGURE 2.13: System diagram for Projected Light 3D Image [12]

The projected patterns are made of very thin light strips, so it is only possible to calculate the deformation of an object in small areas. To avoid this problem, several light patterns are projected, so each of them will cover different areas of the object and other orientations. For this reason this technology is slow and ideal for static objects, but performs 3D modeling with high precision and resolution [12].

## Time of Flight

Time of Flight, or TOF, is a technology that creates three-dimensional images by emitting a laser into an environment, or object, analyzing the time it takes to be reflected. As the name says, the creation of the image is based on the calculation of the time of reflection of the laser, that is, the longer the time, the further the laser traveled. Since the laser travels at the speed of light, laser emission and reception is almost instantaneous, making this technology one of the fastest and most accurate in 3D modeling (figure 2.14).

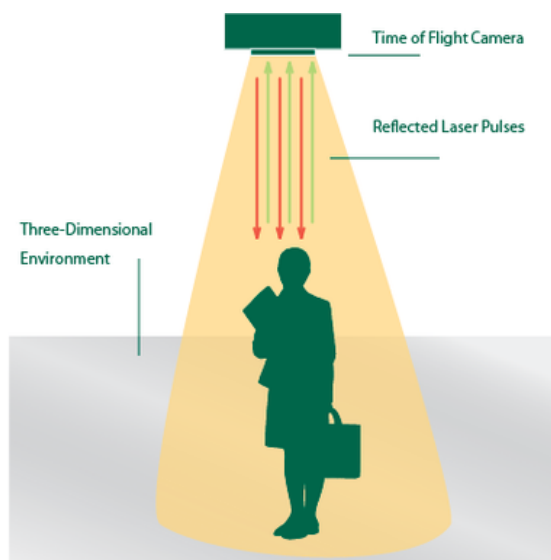


FIGURE 2.14: System diagram for TOF 3D Image [13]

Since this technology allows the generation of more than one hundred 3D images in one second, it is a great approach for tracking moving objects with high precision [13].

## Optical Coherence Tomography

Optical coherence tomography (OCT) is a technology that uses light to capture three-dimensional sub-surface images with micrometer resolution. Like the TOF technology, light is directed at the object, and the reflection time is calculated. But in this case the idea is for the light to penetrate the object to obtain sub-surface images. Hence, this technology has its greatest use in the medicine area, namely as a replacement to ultrasonic or magnetic resonance exams .

Its uses are immense but it possesses greater diversification in the medicine area. It is used in diagnostic medicine, namely in ophthalmology and optometry, where it can

be used to obtain detailed images from the retina, to analyze possible problems in the eye (figure 2.15), and recently in the diagnosis of arterial diseases.

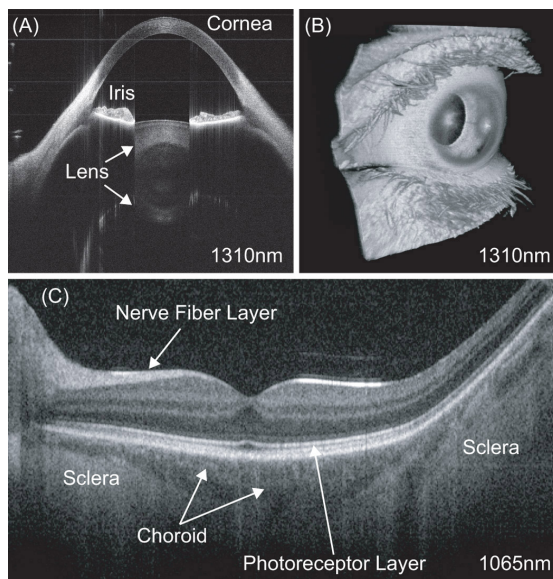


FIGURE 2.15: OCT exam from a eye [14]

As the light used in this technology is in the near-infrared range, to not affect any biological tissue, it limits on image acquisition depth is 1 to 2 mm below the surface of the fabric, because in greater depths the light emitted does not reflect the necessary amount to be detected [50].

## 2.2.4 Type of Image

After analyzing each type of image and seeing the technology behind each implementation necessary, it is possible to see comparisons between the three types of images in the table 2.1. Comparisons like costs, sizes of the sensors, range, among others.

Situations	Visible Images	IR Images	3D Image
Sensor Building	Easy	Easy	Hard
Building Cost	Low	Low	High
Noise Effect	Medium	Medium	Low
Level Applications	Low	Medium	High
Compactness	Low	Low	Low
Quality Image	High	High	Scalable
Range	High	High	Limited
Best Application	Photos	Thermo Photos	3D Scan

TABLE 2.1: Types of image comparison

## 2.2.5 Face Recognition Techniques

The use of a system capable of recognizing someone using their face as an input has many aspects and can be used in many cases, but depending on each case, the input to the system can be different. If the system aims at only surveillance, the images of the face will be obtained from a video, if it is to prevent access to information or location, the input can be static images using 3D or IR cameras.

Some algorithms have already been presented for facial recognition and they depend on the type of data used to acquire the images of the person. These already known techniques for processing and analyzing data can be divided into three categories [15]:

- Video Sequences;
- Intensity Images;
- 3D/IR.

### Video Sequences

The process necessary to recognize a person through videos or images sequences, can be divided into three stages: detection, tracking and recognition. In the detection stage, the person's face is identified, then the face is tracked using some methods that are going to be explained further down (figure 2.16), and that will lead to the recognition of the target person.

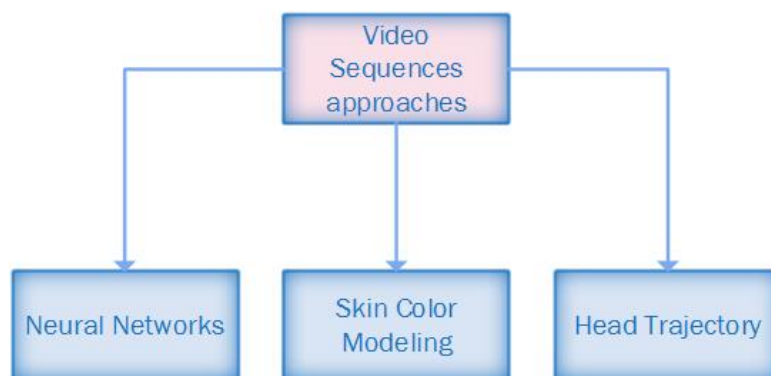


FIGURE 2.16: Video Sequence Methods

#### • Neural Networks

A neural network is "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs." The author of this quote was Dr. Robert Hecht-Nielsen, the first inventor of the first neurocomputer network. In other words, a neural network seeks to achieve an algorithm similar to the human brain. Just as in the brain, the neural network is a set of small processing units (neurons in the human brain) that interact with each other in order to perform a task for which they were trained [51].

A neural network is trained in a way that when it is provided an input, it returns an output based on a previous training. For its creation, it is provided to a program

the inputs, which can be variables and images, and the output(s) for already known inputs, this way the program calculates the mathematical formulas necessary to match the inputs to the outputs.

A system created by Howell and Buxton [15] in 1996, uses as training two sets of images, of the same person. For the association of an image with a person, the system used as training a set of images of the person in a controlled environment (figure 2.17), and another set of images of the person in an uncontrolled environment (figure 2.18). In the first case the person's face was very visible, and in the second case, in some of the frames, they would either record no face or interference due to excess of exterior light.

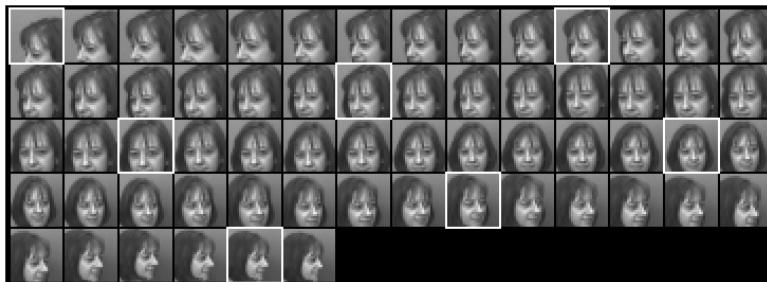


FIGURE 2.17: First set of images, example [15]

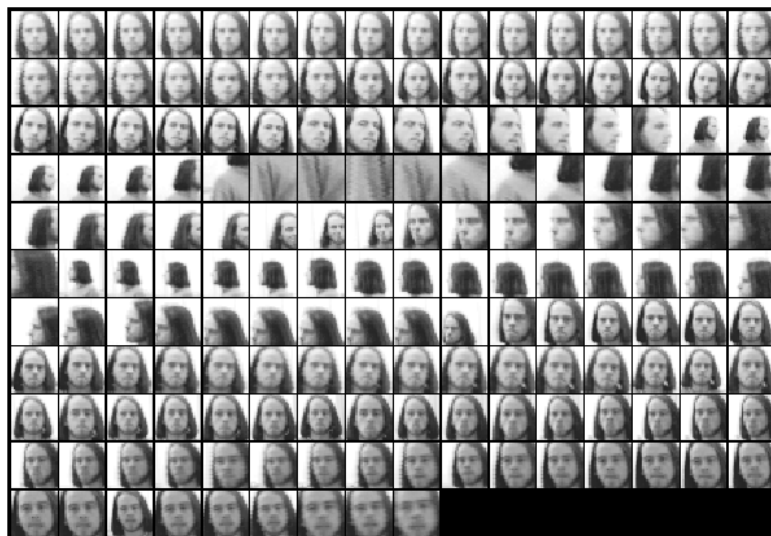


FIGURE 2.18: Second set of images, example [15]

The system demonstrated in 276 tests, when using 278 images for training, a 99% accuracy. When using only 16 images for training, the system only showed a accuracy 67% in 538 tests.



### • Skin Color Modeling

Another system made in Brazil, in 2000, by Campos, Feris and Junior proposes the person recognition by separation of the face, through the division of the face image into eyes, nose and mouth [52].

A database with images of the eyes, noses and mouths of every person to be identified is created. Using as input a video, when in each of its frames the person is identified and the respective smaller images (eyes, nose and mouth) are created (figure 2.19). In a probability system, the images are compared to those of the database, and an array of probabilities with each user in the database is created.

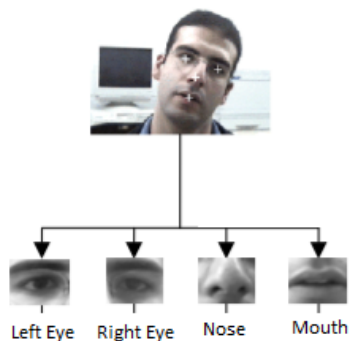


FIGURE 2.19: Facial parts separation

The system demonstrated a high percentage of accuracy, about 97.7%, when used for testing 6 images per person in a total of 29 people.

### • Head Trajectory

A human is a unique being in his own way, since there are no equal people and each one of us has unique characteristics. Some of these characteristics are tics, poses and expressions and in 2001, by the hand of Biuk and Loncaric [16], the proposal of a system that took advantage of one of these characteristics, to recognize a person, appeared.

Throughout our lives we develop poses for everything, when we are sitting, reading a book, or even rolling our heads. It was in this last characteristic that Biuk and Loncaric based the creation of their system, in the trajectory that each person creates when rotating the head, by analyzing a sequence of images that began rotating their heads 90 degrees both ways being recorded (figure 2.20). The process of rotating as well as the trajectory were identified to be used as a comparison on a database with records of the same nature.



FIGURE 2.20: Head rotation record [16]

The tables below present tests performed on the number of point faces that should be chosen to analyze the trajectory, number of frames that the rotation record should have and the accuracy for some of the cases (figure 2.21).



Eigenfaces (n)	Used frames	Recognition rate (%)	Eigenfaces (n)	Used frames	Recognition rate (%)
1	11	85.7	5	6 - 6	67.8
2	11	92.8	5	5 - 7	78.5
3	11	96.4	5	4 - 8	89.2
4	11	100	5	3 - 9	92.8
5	11	100	5	2 - 10	96.4
6	11	100	5	1 - 11	100
27	11	100			

(A) Variation of the face points number

(B) Variation of the Rotation Record frames number

FIGURE 2.21: Accuracy when the number of points or number of frames change [16]

## Intensity Images

Intensity images are digital images resulting from the conversion of an visible light image. The conversion of this image is performed in a matrix, where each position of the matrix represents a pixel of the image, and in each position it is placed a value from 0 to 65535. This value is related to the gray level of the image, where intensity 0 represents black, and 65535 represents the white [53].

In an visible light image, the color of each pixel is converted into the gradients of the primary colors, Red(R), Green(G) and Blue(B). The most direct conversion to gray scale(y) would be the average of the three,  $y = R/3 + G/3 + B/3$ , but as the human eye sees the intensity of each primary colors in different ways, empirical studies show that the best conversion is  $y = 0.3R + 0.6G + 0.1B$  [54].

The existing methods that use this type of input can be seen in the figure 2.22. What each method looks in the intensity image is what sets them apart, because in one case measurements are calculated and in another one representations are sought.

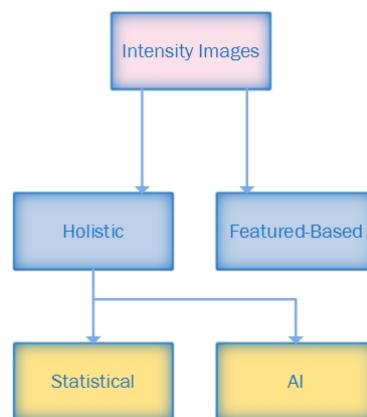


FIGURE 2.22: Intensity images methods

- **Holistic**

One of the approaches to intensity images is Holistic. This approach seeks to identify the faces through global representations, that is, it seeks to recognize a person based on the image and not just biometric characteristics of the person.

This approach can be divided into three groups: statistical, AI (Artificial intelligence) and Multiple Classifier Systems [15]. An example of each approach will be given below.

### Statistical

A very simple holistic statistical approach is to represent the intensity values of a person's image to identify, and make the recognition through a direct comparison of those values with the values of the people in the database (people already known). But this approach has some problems when under the same light conditions, in which the database images were taken do not occur in the images of the people to be identified. Problems like altering the pose or scale of the face can affect recognition too.

In 1991, Turk and Pentland [55] came up with the idea that instead of directly using the intensity face images to make the recognition, using its eigenfaces [17] (figure 2.23) instead, eliminating the problem of illumination and even small variations in the pose or angle when getting the image of the target person. Instead of comparing the intensity images, the comparison would be with the eigenfaces and thus obtaining a probability of equality with the eigenfaces users of a known person database.

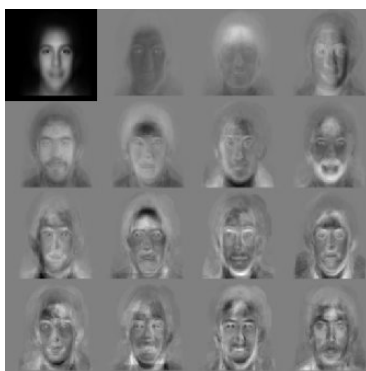


FIGURE 2.23: Eigenfaces Example [17]

### AI

Artificial intelligence, or AI, is a method that uses techniques already referred in the section 2.2.5, where neural networks or machine learning are used as a means to identify the target person. In this case the training is done with several images of identified people, where only one image of the target person is used to make a comparison and that way see if that person is or not recognized by the system.

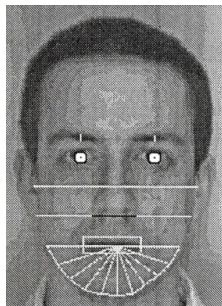
P. Melin, C. Felix, and O. Castillo [15] emerged with a system that could be divided into three neural networks, one with images of mouths, one with images of eyes and one with noses. When the system proceeds to the identification of a person through his intensity image, the image was divided into the three regions previously mentioned and then submitted to the neural networks, expecting a similar output in all.

Their studies have shown that splitting and analyzing each region separately from the face, could increase the accuracy of any system that only uses a neural network for the whole face [15].

- **Featured-Based**

Featured-based relies on identification and extraction to make measurements, of unique characteristics in the face of each person, in the eyes, mouth, nose, among others. These characteristics are extracted and measured, and can establish relationships between them to obtain more unique data. Systems that use this technique, after storing these measurements, compare them with the measurements of people already known to the system, thus making a statistic of appearances, to see if it obtains some high probability.

Brunelli and Poggio, in 1993 [18], designed a system that compared 35 unique measurements on a person's face (figure 2.24) to recognize a person. His results, with a database of 47 people, with 4 images of each person, was 90% accurate, approximately.



---

FIGURE 2.24: Brunelli and Poggio 35 face measurements [18]

### 3D/IR

All techniques seen until now use images of a person in visible light spectrum, but there are some techniques, already implemented, that use other types of images covered in the section 2.2.4. These techniques are very recent and the methods used in each of them are different. In the techniques that use 3D Images, the identification is made by using curvatures that the face has, while in IR images the identification is identical as used in the visible images.

- **3D**

A way of identifying the face of a person is through the curvatures that it makes. The elevation of the chin, curvature that the eyes make, height of the nose, among others. But these features can hardly be calculated with simple 2D images, because 3D coordinates are needed to obtain information like these. The appearance of 3D scan solves this problem, opening up the opportunity of new techniques for facial recognition.

Gaile Gordo, of Harvard University [19], presented a proposal that, using a front image of a face can obtain single points in the image, and with that, perform the person recognition. Using a simple scheme for collecting two points in the image, he analyzed whether the curvature between them was in the positive or negative direction (fig. 2.25). For example, a face point on the left side of the nose, and a another on the right side,

they have a positive curvature, because they have the nose between, and the nose is in a high position than them.

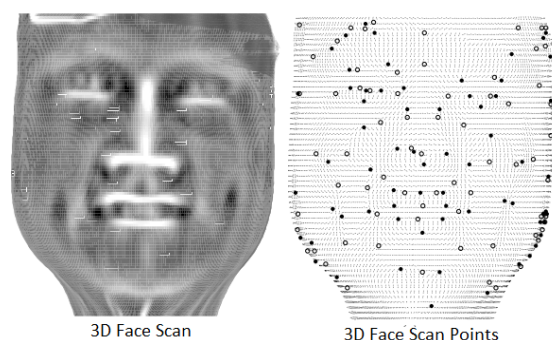


FIGURE 2.25: Gaile Gordo approach using 3D images [19]

With these points, measurements can be made and recognize the "owner" of the face.

- **IR**

Infrared images (Long infrared), as discussed in section 2.2.2, can provide a lot of information regarding temperature differences, and applying it to a human face, it is possible to distinguish veins, or structures/characteristics on the skin. This information is unique to each person, enabling identification with them. But because it is a type of image based on temperature, many external factors can compromise the acquisition of IR images of the same person, like variations in the ambient temperature, the person's mood, nervousness, among others, can originate distinct images of the same person [56].

Even with the possibility of such interference, Wilder, Phillips, Jiang and Wiener in 1996 [57] tested systems using the technique described in section 2.2.5, and others that used the same technique but with IR images. They conclude that the performance of IR systems was not significantly higher than the initial ones, and systems that use visible images are better because they are much easier to acquire than IR images.

## 2.3 State of the Market

Because this is a project to be applied in the automotive area, the research about face recognition implemented so far, has focused in this business area.

The number of implementations found in the market using this type of technology is not very vast. Some of them are still in development or are not yet implemented in their final prototypes, others are not intended to identify the user but the people around the vehicle, but there are cases of final implementations in vehicles that can already be found on our roads.

Something to point out in almost all of the implementations is the fact that the development of the driver identification system was not carried out by only one company, but was the result research and development done by several companies.

Nowadays, it is possible to find a reference to the development of a driver recognition system by Volvo and by BMW. A partnership between Ford, an internationally renowned brand in the automotive industry, and Intel, a company focused more on microprocessor and chipsets development, seeks to produce a system that can identify the

drivers using their face. The companies Peugeot and Citroen, also seek to achieve the same purpose as the others companies already mentioned. Harman, one of the largest companies in the design and development of audiovisual products, refers in it website, a system capable of identifying the driver in order to prevent theft or improper use of that vehicle. The chinese police possess a similar system implemented in their vehicles that is able to identify people with criminal record, next to the vehicle where it is installed. In the following image it is possible to see all companies that are linked to facial recognition systems, either by articles or already existing implementations. The square red represent the companies with projects in development phase, the green one projects already implemented and the yellow, just ideas of final projects.



FIGURE 2.26: Brands/Companies distribution

### 2.3.1 Volvo System

Volvo is a company that likes to be aware of the new available technologies for the improvement of both comfort and safety during the act of driving, thus their interest in a system that recognizes the driver is relatively large[20].

For this, Volvo wants to bet on a system that, using infra-red LEDs, can identify where the driver is putting their attention during driving, as well as their level of fatigue.

The main function of this system is not to identify and recognize the driver, in a sense of safety, not for the implementation of previously made costumizations, but to offer driving assistance, that is, if the driver is tired or distracted, an alert is activated to increase the attention of the same on the road.

The implementation of the sensor would be made on the dashboard of the car in order to facilitate the detection and decrease of occurrence of errors during its operation. The mounting location estimation can be seen in the following image.



FIGURE 2.27: Volvo Face Recognition System [20]

In order to eliminate dependence on an external light to illuminate the driver's face and to illuminate it without causing any discomfort during driving, the system will use infrared illumination and for the acquisition of the image a camera with a infrared filter will be used. This way, the importance of the exterior lighting is reduced and the skin color of the driver does not affect the system at all.

After acquiring the image of the driver, the system will proceed to obtain necessary information, such as the amplitude of the driver's eyes when opened, as well the direction to which they are turned and the angle of the head, making it possible to know where the driver is paying attention, as well as if the the driver is tired or not.

### 2.3.2 BMW

Driver comfort and safety has always been one of the main concerns of companies in the automotive industry, and the BMW company is no exception. With the increase of the "freedom" given to the driver at the suspensions level, where he can control whether a greater or lesser amortization during driving, height and orientation of mirrors, or even change the interior layout of the vehicle, BMW saw a need to facilitate the automatic loading of these changes each time the same driver uses the vehicle.

One of the options that the brand has already sought is the fingerprint identification, which is very practical and it is something unique in each person. But the identification of the driver by the face is even more practical, as it does not involve any effort by the driver.

The concept of the system is very simple, the driver makes the necessary changes, the system will save them and associate them with the driver who is doing so, in a next use of the car by the same driver, the system automatically applies the changes made the last time.

This system, because it involves the identification of the driver, the company still says that, on the use of this system it would be possible to unlock access to the vehicle. The car would only turn on if the driver was in the system data base for authorized drivers, but the confirmation of this feature has not yet been made [21].

For the identification of the driver, an image of the driver would be acquired and then some single measurements would be calculated for each driver. These measurements can be seen in the following image (figure 2.28).



FIGURE 2.28: BMW face recognition system [21]

Some of the measurements should be: the height of the face, from the chin to the tip of the forehead, the length of the face cheekbones, the lateral length of the forehead, the height of the cheekbone from the face to the chin and also the length of the eye. Many measurements could be taken, but the probability of two people having identical measurements is practically null.

### 2.3.3 Ford and Intel

In 2014, the company Intel came up with an idea to implement in cars. As its partnership, Ford was interested in modernizing its vehicles in order to stand out, and was thus interested with the possible implementation of a new system capable of accomplishing something impossible until then [58].

Intel intends to develop a system that, using cameras strategically positioned inside a vehicle, capable of not only identifying gestures made by the driver and associating them with certain functions, but also to identify the driver [59].

Ideas like this had appeared already, so the company decided to innovate even more. Along with the system installed in the car, the car buyer would have access to an application that would connect directly to the vehicle, wherever it was. In this application the buyer could see who was driving the car at the moment, but also set safety parameters, such as maximum speed allowed and whether the driver in question would be allowed to drive the vehicle. If an improper use was identified and detected the driver as not authorized, the photo of the person behind the wheel would be immediately sent to the authorities with a request for apprehension (see figure 2.29).





FIGURE 2.29: Ford and Intel Face Recognition System [22]

This system would innovate in many ways today's automobile market, because in addition to the security already existing in modern vehicles, another one would be implemented, and it would be almost impossible to escape. Cameras used for this task, can also perform the identification of gestures, to implement a new way of interacting with the vehicle.

### 2.3.4 Chinese Police

China is one of the most populous countries in the world, with an estimated population of 1,372 billion people, and it is very difficult to identify possible fugitives of the law in the midst of a vast population. Adding to that fact, the G20 meeting in Hangzhou on September 4 of the last year, the Chinese police had presented at the end of March 2016 a solution to this problem by innovating the methods used in the identification of criminals [23][60].

Developed by a team formed by the partnership between the University of Electronic Science and Technology of China's Police Equipment Joint Research Lab, they were able to develop an extraordinary system. With a 360-degree imaging radius and a range of 60 meters, this system is able to identify, using as a term of comparison a database provided by the Chinese police, criminals or fugitives that are near the police vehicle that contains this equipment. In addition to its magnificent reach and acquisition angle, this system does not need to acquire a complete image of the face of the person to be identified, only needing 3/4 of the face to make a correct identification. One of the great achievements of this team is also the fact that they have managed to make the system small and not very flashy (figure 2.30), to not draw much attention when it goes through a crowd [61].



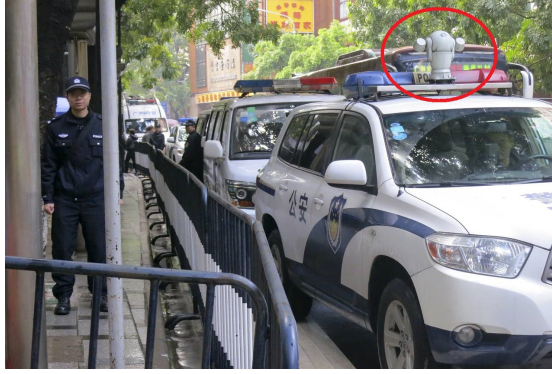


FIGURE 2.30: Chinese Face Recognition System [23]

The system is indeed interesting, making the work of the Chinese police much simpler and safer, so that they can make identifications with a greater security distance without having to put their lives in risk.

In addition to criminal faces, this system also possesses the acquisition and identification of registration plates and types of the vehicles, facilitating the identification and possible seizure of stolen vehicles or identification of vehicles involved in a police fight.

### 2.3.5 Harman

Development of audio-visual and innovative products for the automotive industry has always been a need, and the company Harman was one of those who designed and developed some of these technologies, which can be found in today's vehicles [62].

One of the implementations that can be found by this company is the sound system incorporated in the new Mercedes Benz line, a unique system made for each model of the brand, capable of providing a unique sound quality in the car [63].

In 2013, Harman came up with an article on their blog about an anti-theft system involving the identification of the driver's face. Using a camera installed in the driver's direction, the system would identify the driver through previous training using images previously assigned to the system.

The system, in addition to identifying who is driving the vehicle, would also identify whether or not this person is or not authorized to drive it, alerting the responsible authorities by SMS or MMS. In addition to the alert, the system would also send the GPS coordinates of the vehicle to facilitate its recovery. This system would not only operate during driving but also before the start of the ride, preventing the vehicle from being turned on if necessary, to avoid being used by unauthorized users [64].

### 2.3.6 Peugeot and Citroen

The systems introduced until now identified the driver's face for safety reasons, or even to help the driver in his/her driving, but a team from the Federal Polytechnic School of Lausanne created a system aimed at the safety of the driver himself and those around him.

This team was able to create a system that identifies the face and looks for patterns in order to recognize what emotion the driver is going through at the moment, in other words, if he is nervous, happy or sleepy.

Using an infrared camera behind the steering wheel, the pre-trained system identifies whether the driver is in a state that is too dangerous for himself and for those driving on the road with him, thereby alerting him that it is best to lean on the car and calm or even sleep a little. The figure 2.31 demonstrates the system in action.



FIGURE 2.31: Peugeot face recognition system [24]

However it is still not perfect. The system still suffers from accuracy problems in recognition, in case the driver does not express their feelings according to the training done by the system.

The system still expects improvements such as support for nighttime use, resorting to infrared illumination and a better database of expressions, in order for the system to have better training so it can make better decisions.

The team also said that in addition to the identification of feelings through facial expressions, they also aim to increase the precision of the system through the identification of the lips, making it possible to identify the Driver mood using keywords associated with specific states. The identification of the lips would also be used to improve the hands-free mode, letting the driver make a call without the use of their hands [24].

# Chapter 3

## Methodology and System Specification

In this chapter some technical concepts will be defined for the limitation of the system. Here it will be presented the methodology followed for the system building, as well as the specification of the system and its components. Initially, after an explanation of the methodology, a small overview about the system will be presented, from an independent view and then with its integration into a car. Hardware, such as software, to be used will be presented, along with an expected architecture for the system. Finally, the structure of validation tests will be presented here to, defining what will be tested and how it will be tested.

### 3.1 Methodology

For the structure of this dissertation, waterfall model was chosen as methodology, but with a small change. The waterfall model is linear and sequential, because of this reason it could not be completely adopted in this dissertation, making only a small change in sequentiality between two phases.

Waterfall model is divided into five phases: Analysis, Design, Implementation, Testing and Maintenance [65]. The basic principle of this model is should not advance to the next stage without completing the previous phase, and this principle will be respected. But this model also affirms the non-regression to an earlier phase, making it difficult to obtain some important data to the conclusion of this work. The methodology structure adopted can be seen in the following image.

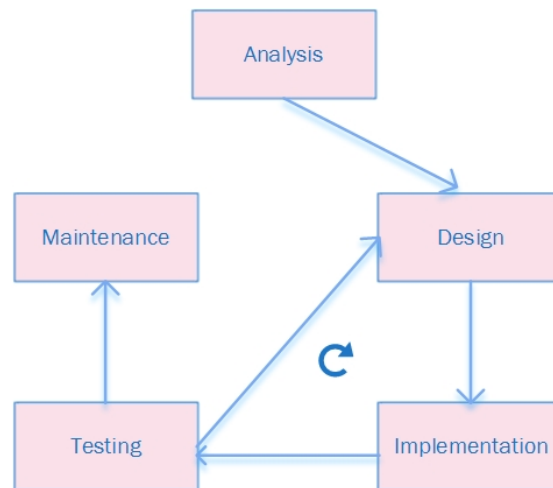


FIGURE 3.1: Adopted Methodology

First phase is the one that will define the follow-up of the entire project (Analysis), because it collects the "cores" needed for the project. In this stage the requirements, objectives, needs and the reason for the realization of the project are defined. Also at this stage are defined the project planning, how it will be carried out, when it will be carried out and what results are expected at its conclusion.

Next phase is Design. Here it is specified the technical questions of the project, the hardware to be used, the software and the chosen approaches. Also in this phase it is described the overview of the final system and how it will interact with other systems, and how they will interact their components, communication protocols, connections, among others.

The third phase is the practical phase (Implementation), where what has been described in the Design is implemented. If an algorithm was developed, it will be presented in this stage, divided into small sections for better understanding. Small tests can also be performed here to prevent future failures. The hardware integration defined in the previous phase will also be presented, as well the software associated to the integration.

Testing phase is the realization of tests for validation purposes of the developed system. Here it is presented data about the system. Its sensitivity, precision, possible errors will be described as well, to check if the requirements described in the first phase are fully filled. As only in this phase the failures of the system will be detected, the retrocession to the Design phase, to correct them, can occur.

Final phase is Maintenance, where the conclusions taken from the system will be presented. Since it is the last phase, the system will be analyzed again to detect possible errors or failures that were not detected or could not be resolved in the previous phase, as well as presenting possible solutions for the correction of these faults.

This methodology allows a very useful division for the system manufacture, defining very concise deadlines and times for the accomplishment of each task or objective. Since this methodology allows the regression of the Testing phase to the Implementation phase, in the event of any detected error, the system can re-enter in the Implementation phase for its correction, reducing the probability of detecting any failure in the Maintenance phase.

## 3.2 System Requirements

Before defining the system architecture, some requirements have to be defined. Some of them have already been presented in the section 1.5, but will again be presented in this section in more detail. The requirements of the system can be divided into two categories:

### 3.2.1 Functional Requirements

- Collect images from a camera;
- Process this image and identify/recognize faces;
- Enable driver identification even if multiple faces are captured by the camera;
- Ensure 75% rate of driver recognition;
- Enable the initiation of an driver recognition through a command received by the server;
- Communicate the face identity to a server using the TCP/IP protocol suit.

### 3.2.2 Non-functional Requirements

- Enable the identification of several users (not at the same time);
- No camera calibration required;
- Identify the driver in just over 1 second;
- Create a prototype that can be integrated and tested in a "Real Life Driving Simulator".

## 3.3 System Overview

As already mentioned in the first chapter, the system to be developed has the objective of the sharing of the results with the project "INNOVCAR: The Cockpit of the Future", resulting from a partnership between Bosch and Minho University. From this partnership also came the DSM (Driver Simulator Mockup), a real life car simulator, built with technologies in development. It is also expected from the system developed in this dissertation, a communication with DSM, so that the DSM can load the profile associated to the driver. A general overview of the system can be seen in the following diagram.

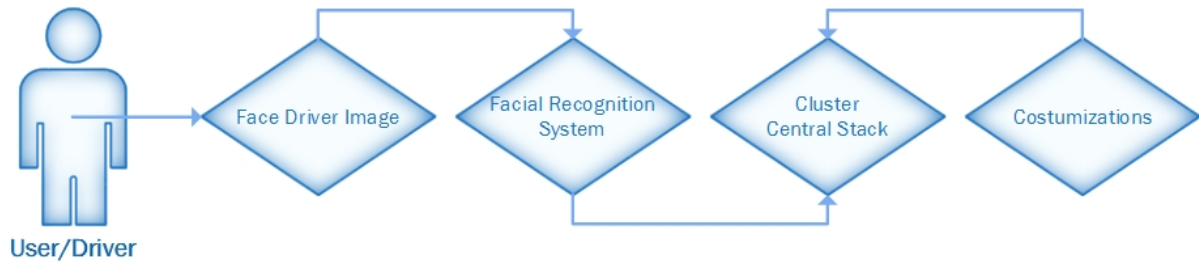


FIGURE 3.2: System Overview

As the diagram presents, after collecting the face user/driver image, it will proceed towards the recognition of the driver to inform the Cluster, or Central Stack, of the driver's identity and thus load their specific customizations.

Cluster is the dashboard behind the steering wheel, where it is possible to see a speedometer, rotations meter, and fuel meter. Central Stack is the panel located between the driver and front passenger, where it usually has the controllers to radio, GPS, air conditioning, among others.

### 3.3.1 Loading System Overview

In order to perform the system shown in fig.3.2, some components and structures are required to be able to perform the profile loading associated with a user. In the diagram, figure 3.3, it is possible to see these needs.

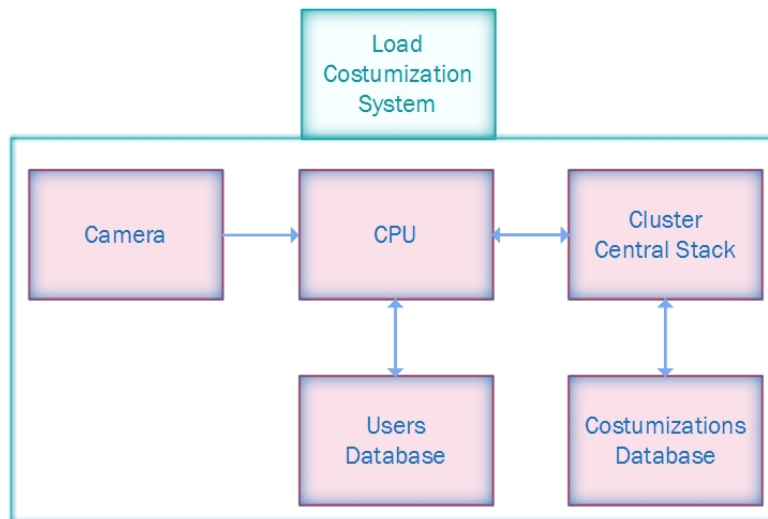


FIGURE 3.3: System Overview

As it is possible to analyze in the above diagram, the main components to the realization of this system are: the camera to acquire images, the CPU to process the images, Cluster/Central Stack to make the loading and the databases to store all the information about the users. Currently two of these components already exist, the cluster is already implemented and operational, along with the customizations database.

Defining the development of the remaining components, and what components are those that will perform the driver recognition, is the main objective of this dissertation.

### 3.3.2 Facial Recognition System

As already mentioned in the previous section (3.3.1), the components for performing driver recognition are the Camera, CPU and Users Database. Since they form the Facial Recognition System, in the diagram shown in fig.3.4 it is possible to see in more detail the interactions between each other, as well as the communication with the DSM (Cluster/Central Stack) and their functionalities.

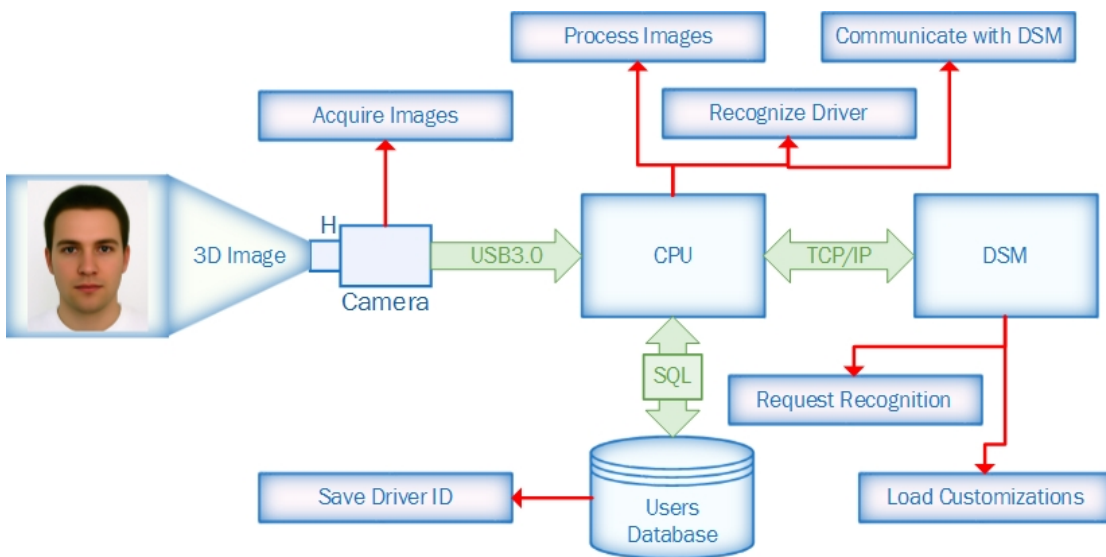


FIGURE 3.4: Complete Facial Recognition System

In the above presented diagram it is possible to see all the functionalities of each system component and the communication protocols.

After a 3D image is captured, the image will be processed by the CPU. The type of image collected, as well as the communication protocol that the camera uses, will be explained in the following sections. After processing the image, the CPU will analyze the contents of the database to verify if the driver captured on the image is already registered, and send that information to the DSM. If it is a new user, the data will be stored in the database, associated with a name.

This entire process will be triggered by the DSM, who will send a request to identify the driver, so that way, he can receive his ID and load the correspondent customizations.

The protocol between the DSM and the CPU has not been defined, but imposed by Minho University and Bosch, because the current systems installed in DSM uses the protocol TCP/IP to communicate, so, for the CPU to communicate with the Cluster, or central Stack, it must also use the same protocol.

## 3.4 Technology Specification

For the development of the final system to this dissertation, it is necessary to choose the type of image that will be collected from the driver's face (consult section 2.2), in

order to perform the recognition. This will also determine which camera to choose, and what technique to use.

Some factors have to be taken into consideration when selecting the image type. One of the main criteria is the availability of cameras that can acquire this type of image, because this will define the time necessary for the acquisition of the equipment, and lead to delaying the conclusion of this dissertation. Another reason of extreme importance is also the time required to develop any algorithm for image processing, for the same reason as the previous criterion.

### 3.4.1 Technology Criteria

The criteria to choose the type of image to acquire are:

- Cameras available in the market;
- Time needed for development;
- Range;
- Implementation costs;
- Implementation size;
- Variations on environment temperature;
- Variations on illumination;
- Image quality;
- Others possible features;
- Available support.

### 3.4.2 Technology Definition

Using the information described in section 2.2, along with an internet search about the existence of camera types and a search for support sites for those who want to develop algorithms capable of recognizing people, by their image face, it is possible to present the table 3.1, with the response of each image type, for each of the criteria described in the section 3.4.1.

Criteria	Visible Images	IR Images	3D Image
Cameras available in the market	High	High	Medium
Time needed for development	High	High	Medium
Range	High	High	Medium
Cost of implementation	Low	Low	Medium
Implementation size	Low	Low	Low
Variations on environment temperature	Low	High	Low
Variations on illumination	High	Low	Low
Image quality	Medium	Medium	High
Features	Low	Low	High
Available support	High	High	High

TABLE 3.1: Type Images Comparison



As the table shows, regarding the two main criteria, availability of cameras that can acquire the type of that image, and time needed to develop an algorithm of facial recognition, it is possible to verify that the type of 3D image has the advantage, as there are not many cameras on the market, but the existence of support forums reduces the time needed to develop an algorithm for facial recognition. Although it has a higher cost of implementation than the other types, 3D image wins in the tolerance to variations of the ambient temperature and lighting, as well in the quality of the images. So, the best choice of the technology used to acquire an image of the driver's face is 3D.

## 3.5 Hardware Specification

Since the image type is defined in the previous section, it is necessary to select the camera for image acquisition. The next subsections will approach the cameras that can acquire the type the images, and have a small size, the criteria for choosing one of them and the chosen one.

### 3.5.1 3D Cameras

The amount of cameras that can be found in the world market is very high, but those that can only be incorporated in a car, are not so many. Some factors have to be taken into account when it comes to choosing a camera to be used in the automotive world. Size is the main feature, but its cost also has to be taken into account. In order to make the choice of a camera for performing facial recognition, in a car, using 3D images, the following criteria were chosen:

- Size;
- Range;
- Cost;
- Robust build;
- Resolution;
- Frame number(FPS);
- Communication type;
- Market availability;
- Technical support.

In the next step, a market study will be made on the possibilities that can be found in today's market, comparing them using the above criteria described, in order to make the most appropriate and reliable choice.

The choice of the camera will not only define the technique that will be used for facial recognition, but will also define the technical characteristics of the CPU (diagram fig.3.4). Some of the 3D camera that will be referred, need the recent generation of CPU, another one doesn't need a recent version and the last one does not even need CPU, because it already comes with it incorporated.

CPU requirement also determines the size of the system to develop, so the size criteria takes into account not only the size of the camera, but also the CPU required for processing the acquired images.

## Market Study

In the current market, there are three cameras capable of acquiring a 3D image and still maintain a compact size, not compromising the quality of the image. The cameras found will be presented below.

- **IntelSense SR300**

The IntelSense SR300 camera (figure 3.5) is the latest version of the camera, still existing, F200. It is a version for developers that aims to test it in applications that intend to use 3D depth data. It is a very compact camera that allows the acquisition of images of high resolution at a rate of 30 to 60 frames per second. The company Intel created a forum only for those who are developing projects using this type of cameras. It is possible to obtain support from engineers of the company, or from other users of the forum.



---

FIGURE 3.5: Camera IntelSense SR300 [25]

- **Kinect**

The Kinect camera (figure 3.6) is a commonly known camera in the current market. Introduced for the first time in 2010, it was a camera initially designed for the Xbox 360 console, from Microsoft, but was also used later by Sony, on on the PS3 console. Its main use in these consoles is the body tracking of the player, so that the player can play without needing a controller. Currently it can also be found on the Wii console, made by the company Nintendo.

As it is a camera already well known by the public, many face recognition, face tracking or 3D scan projects were developed using this camera.



---

FIGURE 3.6: Camera Kinect [26]

- **mvBlueSIRIUS**

The mvBlueSIRIUS camera is the most expensive of all the cameras mentioned here, but also the one with more features and optimizations. Using a technology that the company Matrix Vision, the manufacturer, calls 6D, it is capable of not only obtaining the 3D points coordinates of the image it acquires, but also the vectors of motion of each point.

Its optimization and compression is so high that the components necessary to process the images are in the same box (figure 3.7), and the occupying image processing only

uses 5% CPU processing capacity. The user can use the remaining 95% on whatever he/she want.

Its construction was initially intended for industrial use in controlling pass-through items, but its ability to acquire 3D data enables it use for face recognition.



FIGURE 3.7: Camera mvBlueSIRIUS [27]

### 3.5.2 Camera Definition

The comparison criteria for the cameras have already been defined in the section 3.5.1, and the result of each camera for each criteria can be seen in the table 3.2. All data in the table was the result of searches on forums, company information, or results of sent email to the camera manufacturer.

Criteria	IntelSense SR300	Kinect	mvBlueSIRIUS
Size(length)(mm)	110	249	234(with CPU)
Range(mm)	200 to 1500	400 to 4000	800 to 2500
Cost(Dollar)	149	100	Unknown
Robust Build	High	Medium	High
Resolution	720p 1080p	720p 1080p	1080p
Frame Number(fps)	60 30	30	25
Communication Type	USB3.0	USB2.0	Unknown
Market Availability	Medium	High	None
Brand Support	High	Medium	Medium

TABLE 3.2: Comparison between cameras

Using the table presented, the best camera for performing face tracking is the IntelSense SR300, because it is the one that best fits the established criteria, firstly because the camera mvBlueSIRIUS has not yet been released (only in 2017 will it be possible to acquire this camera, the body of kinect is made of plastic, which does not give it great protection) and also has a considerable length. The IntelSense camera has a small length and is capable of collecting 60fps (frames per second) with a resolution of 720p.

The fact that the SR300 uses the 3.0 version of the connector Universal Serial Bus (USB), that can allows a data transfer of 5GB (Gigabytes) per second (theoretically number), reducing any delay between the image capture and image transfer. As it is a camera with recent origin, it requires some high CPU characteristics, which are:

- OS Supported: Microsoft Windows 10;

- Recommended CPU: Intel 6th generation processor or above;
- Recommended RAM: 4 Gigabytes.

This means that the chosen PC must support the OS (Operative System) Windows 10, and be constituted by an Intel processor of the sixth generation, because the existing drivers for the camera were only developed for this architecture.

### 3.5.3 CPU Handling

With the camera chosen, the choice of CPU to process the data is determined. Since in the section 3.5.2 some technical characteristics have already been defined for the CPU, only one product on the market fulfills these requirements, and manages to maintain a small size. This product is the Intel NUC 6th Generation, also produced by the company Intel, and can fulfill the requirements while maintaining the size of 115mm length and 111mm wide.



FIGURE 3.8: Mini PC-Intel NUC Kit [28]

As it indicates, it is a mini PC (Personal Computer), and was created with the intention of replacing large setups necessary to get a good PC. It is a mini PC where you can install a Solid State Drive (SSD), which allows the boot (startup) of the OS in seconds, making it possible to start any system that uses it in a few seconds.

This CPU has an Ethernet plug, thus allowing the connection, using the TCP/IP communication protocol, with the server, and has four USB3.0 ports, which allows for the camera plugin and transfer data on the total usage that the connector allows. The RAM that can be installed is expandable to 32GB, and since it contains the Intel i5-6260U processor, which is the sixth generation, the OS Windows 10 can be installed.

The only problem with this PC is that it needs 19V of power supply, and typically a car battery supplies 12V, so in order to use this CPU in a car, a 19V to 12V converter would be needed.

## 3.6 Software Specification

For the development of this dissertation, some programming languages have to be selected to implement the necessary algorithms. To select them, some criteria have to be defined, and is the following list:

- Language flexibility;

- Learning curve;
- Previous experience;
- Developer support;
- Developer community;
- Camera support.

### 3.6.1 Programming Languages Selection

Only three languages support the chosen camera, C, C++ and C#. Of these three, only two will be chosen, C# for the creation of the support algorithm for the camera and C++, for the server that will establish the TCP/IP connection. Both have plenty of support from other developers and there is a large support community for anyone interested in using these languages.

### 3.6.2 Software Development Platforms

With the choice of the programming languages, it is necessary to choose the platforms for the algorithm development. There are many choices, but the language chosen already limits many options, but still needed criteria are necessary to make the choice:

- Developer support;
- Camera support;
- Developer community;
- Functionality;
- Cost;
- Code portability;
- Previous experience.

### 3.6.3 Development Platforms Selection

Using the list of criteria defined in the previous section, along with a search of what can be found in the market, the list of platforms chosen can be seen next.

#### Visual Studio

For the development of the algorithm in C# for the camera, the Visual Studio framework was selected. Not only will it create the images processing algorithm received, but also the TCP/IP connection between the system and the server, and communication with the users database.

The selection was due in large part to the fact that it is the only framework that IntelSense offers support when it comes to the development of algorithms for the use of its cameras.

## Matlab

There are some algorithms already in commercialization that allow users to perform facial recognition, only requiring the user to introduce some images of the one to recognize, or enter an image for the program to identify faces.

These algorithms only allow its use in executable form, and thus do not give access to any source code. The Matlab program was selected for execution of those algorithms.

## QT Creator

The framework selected for the development of the server that will initiate the facial recognition remotely, was the QT Creator. The choice was due in large part to the work experience already acquired before the beginning of this dissertation, and by the existence of a large supportive community for those who use it.

## DB Browser

A final program is necessary, only for debugging and analysis, in order to verify that all the necessary information is being correctly saved in the users database. The DB Browser program has been chosen, and will only be used to verify what information is stored in the database, and how it is stored.

### 3.6.4 Algorithms Libraries

For the development of test algorithms, or algorithms for the selected camera, some libraries are required that contain functions to develop them. Two libraries were selected to respond to this need, the Intellisense library and the OpenCV library.

#### OpenCV

It is possible to find several algorithms of face tracking, eye tracking, hand recognitions and others, using the functions of this open source library.

This library was only used for performance tests, specifically, how many frames could be achieved using this library as a means to process the images of a camera. The tests performed showed very low values, in the order of 15 to 19 frames, so, this library was used only in the tests and not during the development of the system of this dissertation.

#### Intellisense

Intellisense is a library by Intel developed to do/enable the connection to the IntelSense gama cameras. As the selection camera is one of them, the use of this library is essential, and without it, it would not be possible any use/manipulation of the chosen camera.

These libraries come with algorithms that implement already some features, such as facial tracking, voice recognition, eye tracking and even hand gesture recognition.

As mentioned in section 3.5.1 Intel created a forum, so that those who bought their cameras could share problems, or developments, with other users, but this forum also aims to help and support those who are using the Intellisense library.

## 3.7 System Architecture

For the implementation of the system to be developed, it is necessary to define and design all the parts that compose it.

For the total system design, it is indispensable to take into account the material to be used, so as to define which communication protocols will be used between them, and the software for the development of the necessary algorithms, since it is necessary in order to design these algorithms.

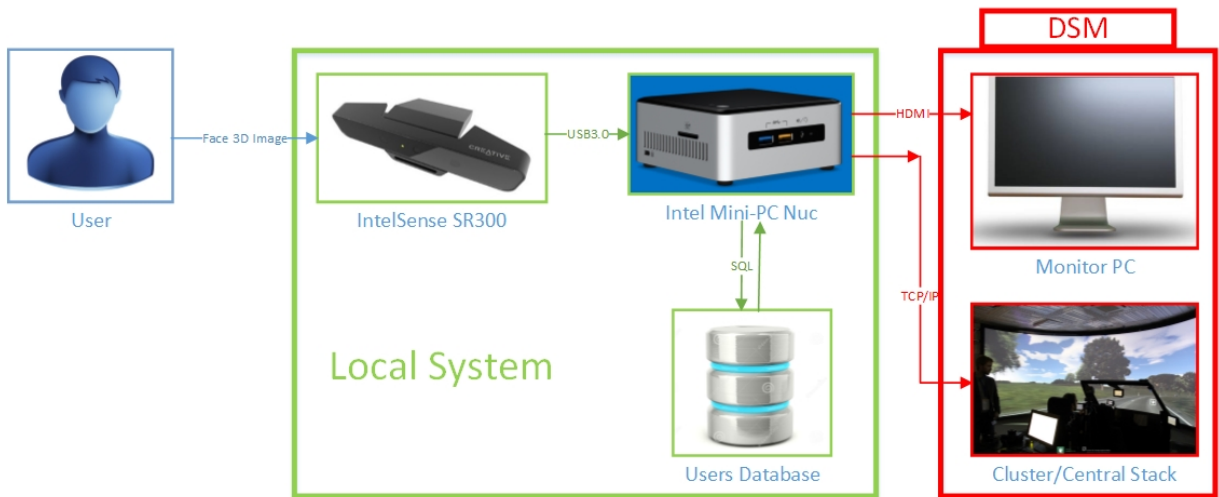


FIGURE 3.9: System Design

In the figure 3.9, it is possible to see the various components intended to compose the final system, the communication protocols between them, and also how the algorithms to be developed for the system will be divided.

The system is constituted by the local system, which will be the core of the system, and by DSM, which will consist in two algorithms, by the server, which will replace the cluster/central stack function, and by the GUI (User Graphical Interface) that will be a method of monitoring, or activating, the local system.

### 3.7.1 Local System

As stated in the previous section, the local system will consist on three parts, and this section will address the core, the local system. With the hardware defined in previous sections, it is possible to now see in the figure 3.9 the three components that compose it. The algorithms to be developed in this part of the system will run on the Intel Mini-PC NUC component.

For a better understanding of the algorithms to be developed for this part, a series of flowcharts will be presented, with the proper explanation for each one.

## Main

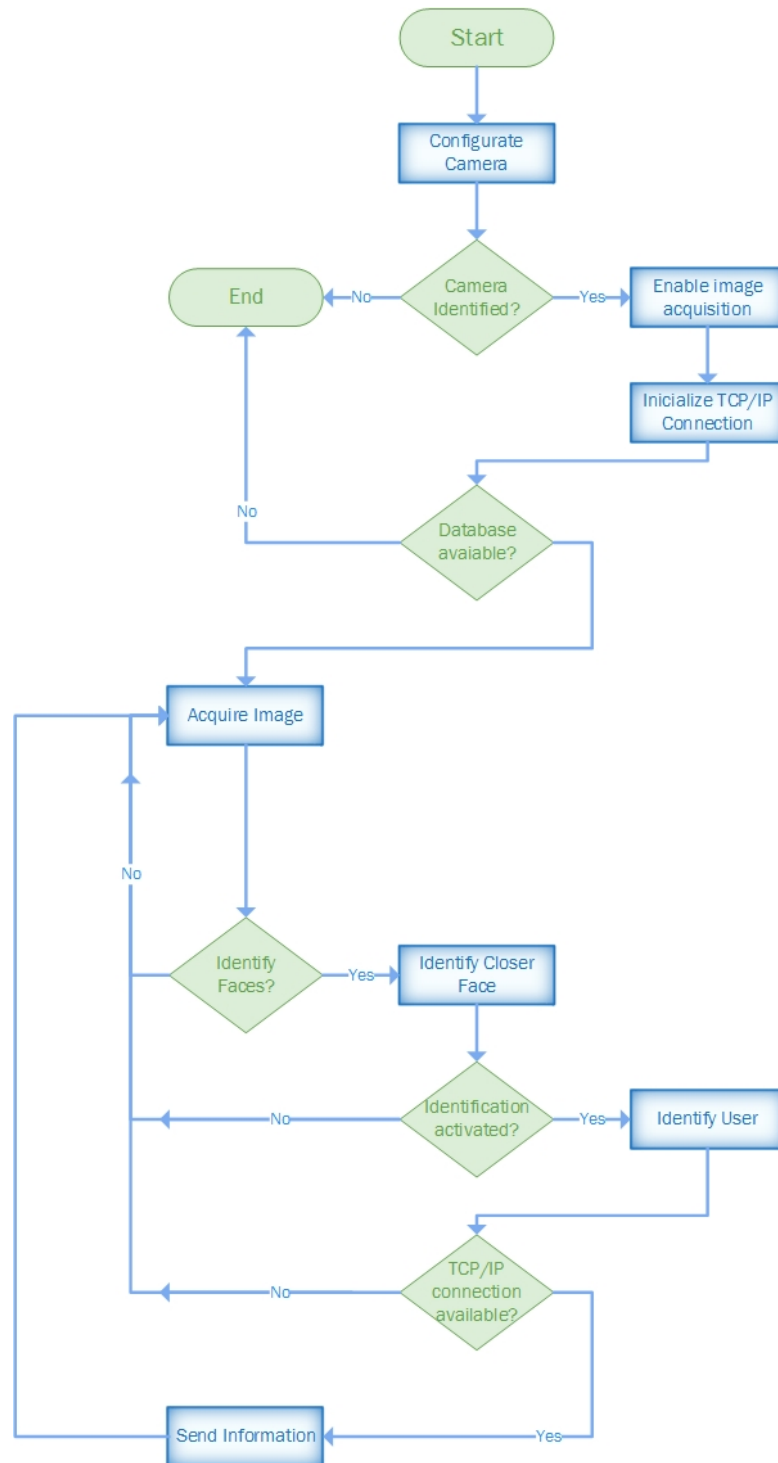


FIGURE 3.10: Main Local System Flowchart

As it is possible to identify in the flowchart 3.10, initially it will be necessary the configuration of the camera, so that it can obtain the 3D images, and for the system to



know if the camera is even connected. The server interface will be made immediately after this, followed by the system entry in a cycle where it acquires user images to perform the recognition.

In this flowchart the identification of the user closest to the camera is represented only by a block, despite it being a more complex process, with the method to identify the user also being represented here by a block. In the following sections, they will be specified in more detail.

## Multi Face

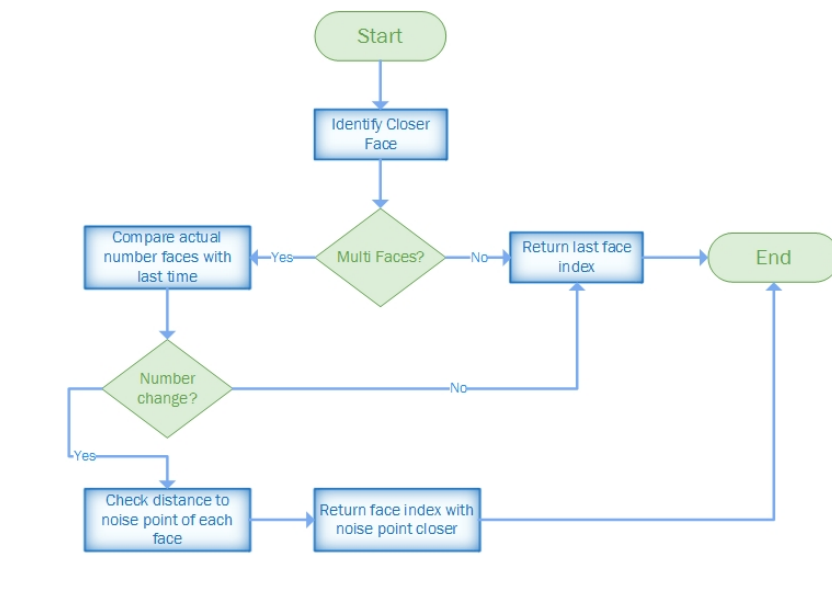


FIGURE 3.11: Closest User Identification Flowchart

The library already mentioned in the section 3.6.4 contains a feature where, by providing an image, it is able to identify specific points on the user's face (landmarks) and if there is more than one face in the image. Using these features as a base of development, it is intended to develop an algorithm that uses the  $Z$  coordinate of the point at the nose tip, and with the number of faces in the image, identify the face with the point of the nose with the lowest value, and that way, if recognition is requested, it will be the user that will be used as input to the system.

As mentioned in the flowchart 3.11, this function will only be executed each time the number of faces identified changes, preventing information from multiple users being used during a recognition.

## Face Position

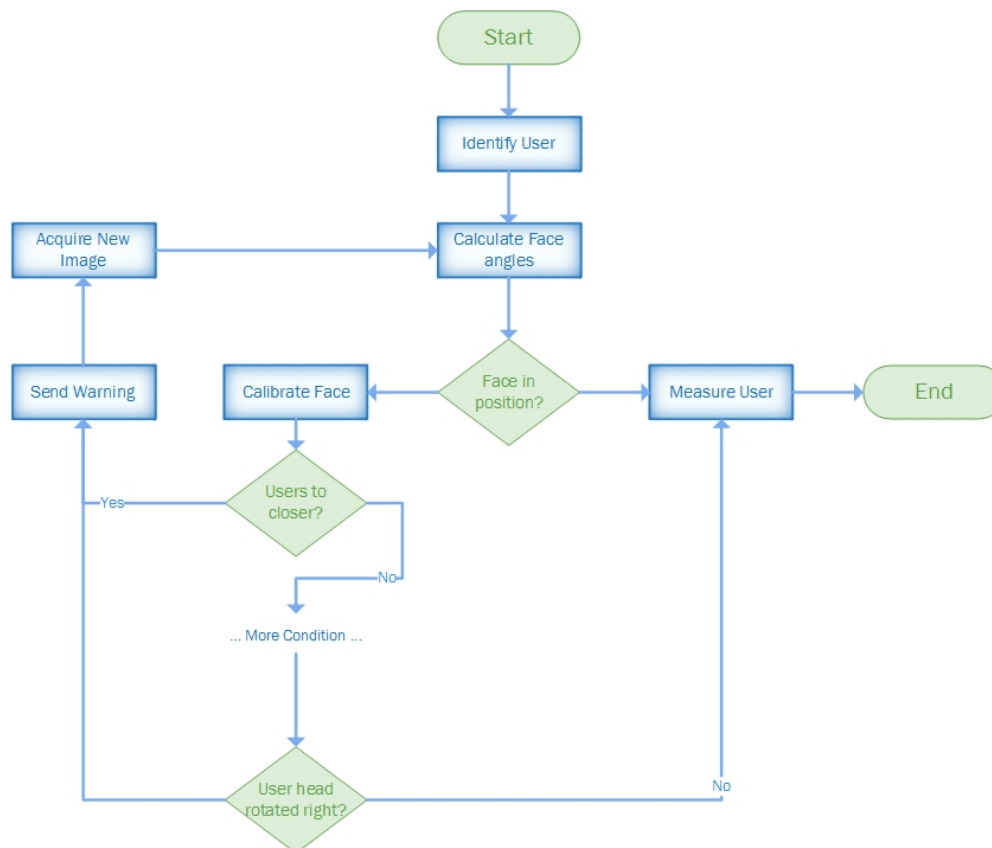


FIGURE 3.12: Face Calibration Flowchart

Using as reference tests already made to the chosen camera (see section 3.5.1), the 3D coordinates of the landmarks of a user's face are very unreliable due to the distance the user is from the camera.

As was explained in the section 3.7.1, the 3D coordinates of user landmarks will be used as system input for recognition. To prevent the coordinates from changing for the same user, the driver must first follow a series of instructions to be in an specific position. In the flowchart 3.12, it is possible to see what will be taken into account to have the user in the desired position, namely characteristics such as distance to the camera, face direction and rotation, among others.

## Face measures

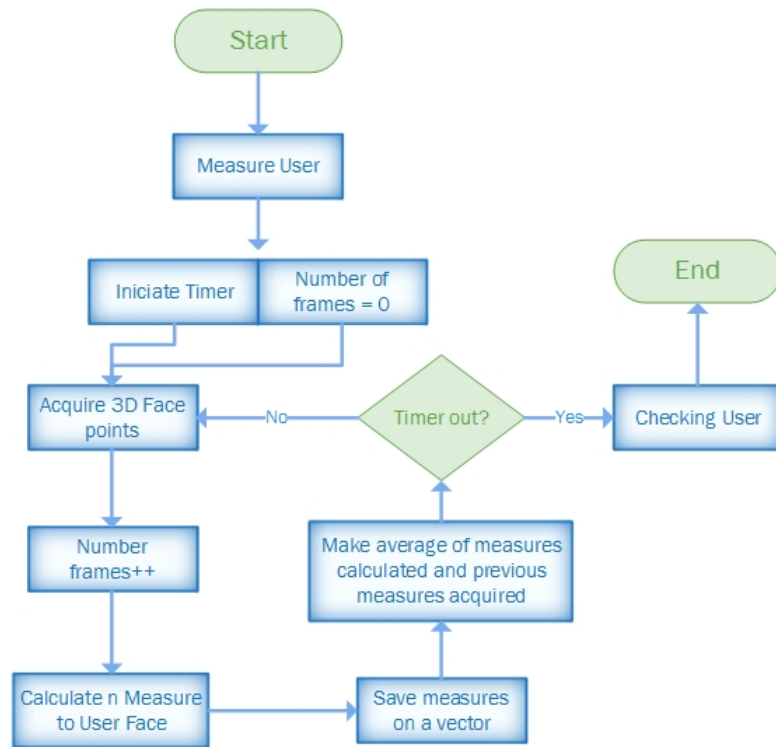


FIGURE 3.13: Face Measurements Flowchart

Using one of the features that the Intellisense library provides (see section 3.6.4) through the 3D coordinates of the user's face landmarks to recognize, measurements will be taken. The development of the algorithm for the acquisition of the points was made by Intel, but the design of the algorithm to be developed to perform the measurements can be seen from the 3.13 flowchart.

In this flowchart, it is possible to see that for the measurements a timer and a counter are necessary. As already mentioned in the previous section, the 3D points are very unstable, and in order to reduce this instability, points will be acquired, the measurements calculated and then made a average. For this it is necessary a timer to stop obtaining the points and a counter to know how many measurements/number of images have been used.

After the measurements are made, and the averages calculated, a user checking is performed on database (block 'Checking User').

## Face Checking

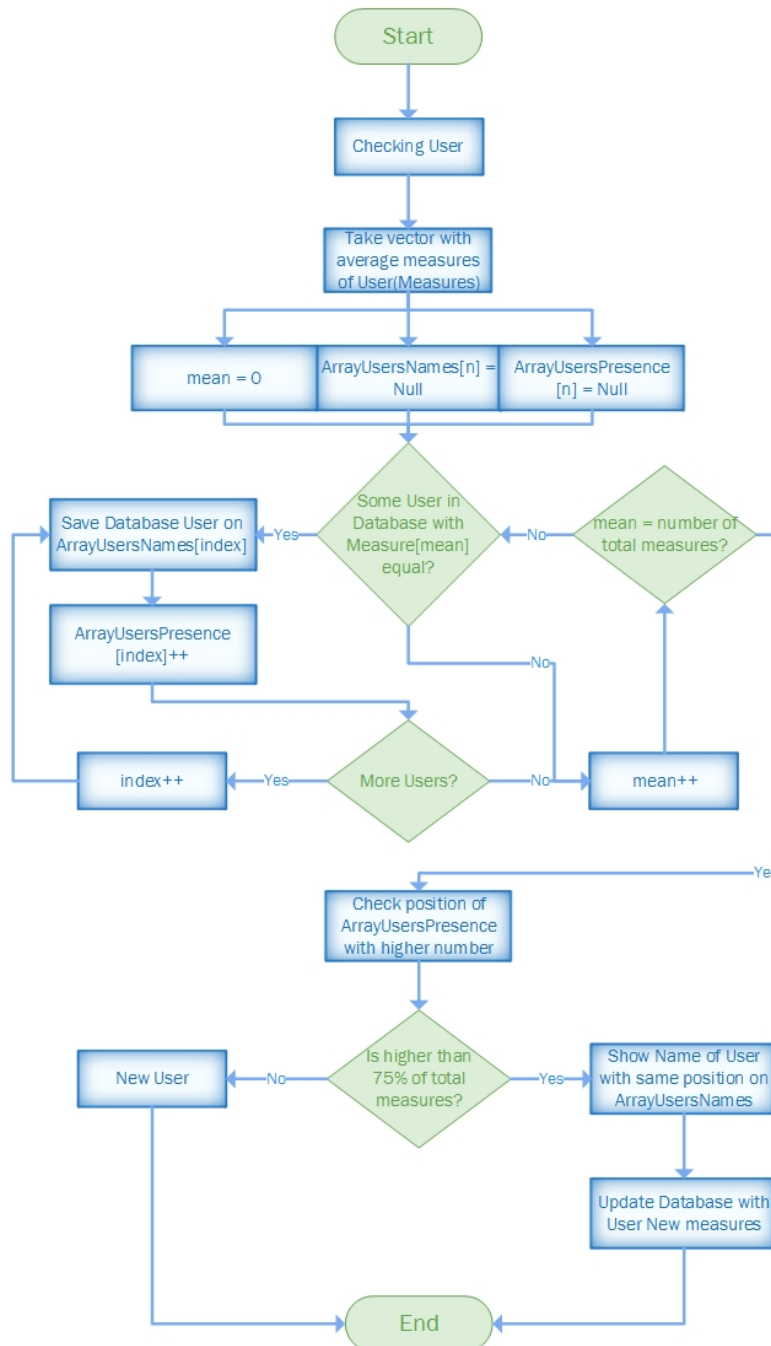


FIGURE 3.14: Face Checking Flowchart

For the driver recognition, an identical output to the neural networks (see section 2.2.5) was defined, not being made recognition by means of an absolute equality in all the measurements, but only in a percentage of them.

With a vector containing the average of the driver's face measurements, it is verified again if the database is active. For each measurement of the driver, it is verified if there is a correspondence to a user in the database (flowchart 3.14). Each time a user

is compatible, its name is stored in an array, position X, and in an array that registers the number of compatibilities, in its position X the value is incremented. So if in measurement zero, user John and Tommy have an identical measurement, ArrayUsersNames array will have in position zero John and Tommy in position 1. After that in ArrayUsersPresence is registered in position zero the number of compatibilities that user John has with the driver and in position 1 the number of compatibilities that user Tommy has with the driver.

As in the neural network, instead of a direct answer stating that the driver is a certain user, it will obtain a number of compatibilities of database users with the driver, a kind of likeness of appearance. If this probability is high ( $>75\%$ ), then the driver is most likely that user.

Over the years, the human face tends to change gradually, that is, if the system calculates the measurements once, and never more, a grown beard, for example, would affect the recognition because it altered some of the measures, so the system, each time it recognizes the driver, updates the database with the new measurements to avoid these situations.

### New User

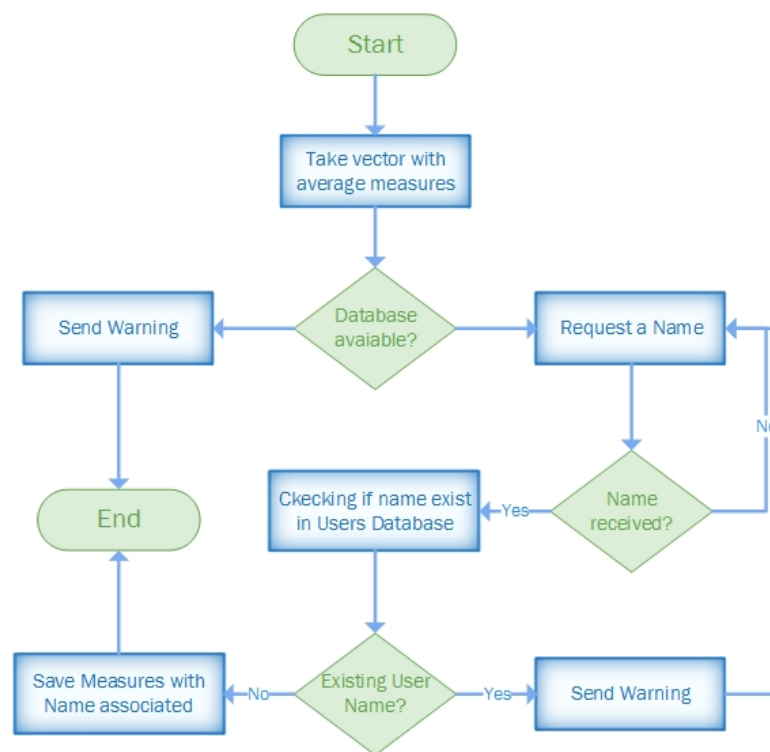


FIGURE 3.15: Recognition of a new Driver Flowchart

In the event that none of the compatibilities is greater than  $75\%$ , it could mean that the driver is still unknown. In this case the process needed to save the information about this new user is very simple.

As can be seen in flowchart 3.15, it is only necessary to acquire a name for the association of previously acquired measurements in the database. The name can be

acquired by the server, or by the GUI, as it will be possible to see in the following sections. The given name will only be analyzed first to check if there already is a user with the same name. If it does not exist, the measurements are saved and a signal is sent stating this.

### 3.7.2 Server System

As already mentioned, the local system is the part of the system with major importance, but without the server its usefulness and practically is null.

The final system aims to communicate with DSM (see section 3.3) using TCP/IP, but it is always available for for testing for someone with authorization. The server system responds to this need, simulating an action that DSM would take, and also test the TCP/IP connection.

For this connection, it is necessary to define two important parts of the server system, the communication protocol, which will define the messages to be sent, the responses, and also the architecture of the system, how it will handle a connection, and what it does in each of them.

#### Communication Protocol

A communication protocol is a set of rules that are required to comply by all programs or machines that intervene in a data communication. Without it, communication would be chaotic and, therefore, impossible.

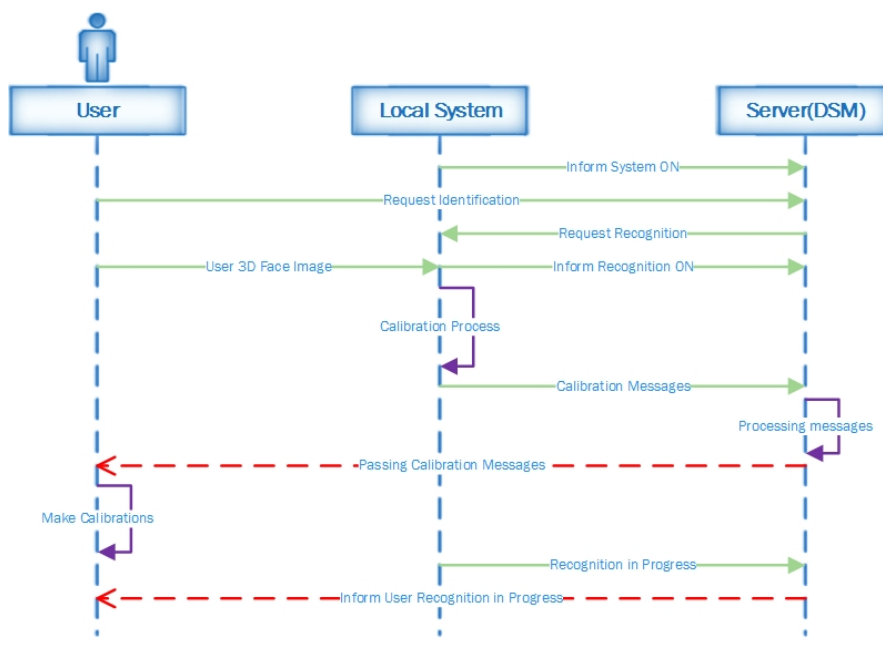


FIGURE 3.16: Start Recognition Sequence Diagram

In this case, diagram 3.16, it is possible to see the entire communication starts with the local system informing the server that it is operational, so when the driver requests for a recognition, the server will have the information if the system is ready to perform the recognition or not.

After the recognition request has been made, local system sends a set of messages to the server, and then the server to the user, so that he/she corrects his/her position. Once the calibrations are done in the user's position, the recognition is done and in this case the communication can enter into two situations.

- **First Case**

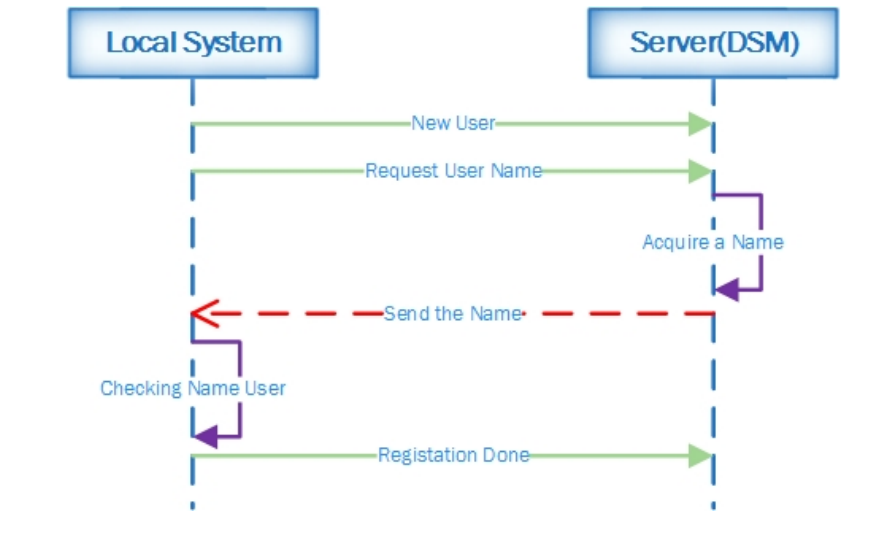


FIGURE 3.17: Good Name Sequence Diagram

In the first case, diagram 3.17, we have the situation of the server supplying a valid name to associate the measurements acquired from the driver's face, in the users database.

It is a very simple process, and after the local system verifies that the name is valid, it sends a message to inform that the registration was carried out successfully.

- Second Case

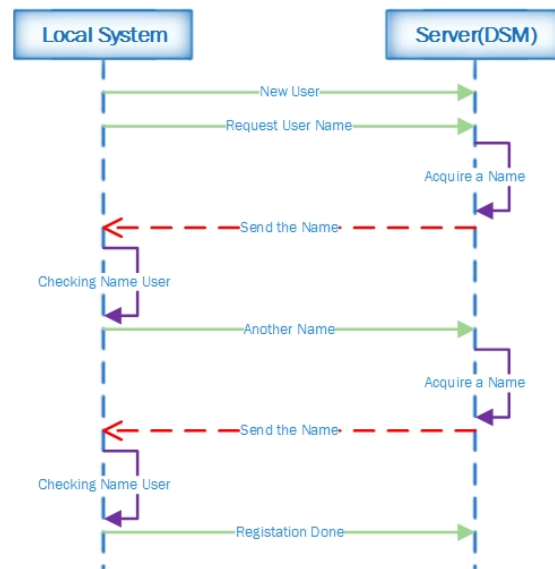


FIGURE 3.18: Error Name Sequence Diagram

In the second case, diagram 3.18, we have the opposite case, where name is not valid, so local system makes a new request for a new name.

The protocol in this case is to repeat the process of acquiring the name in the first case, until the server supplies a valid name. There is no limit to the number of validations, although there may be some restriction on the side of the server.

### Server System Architecture

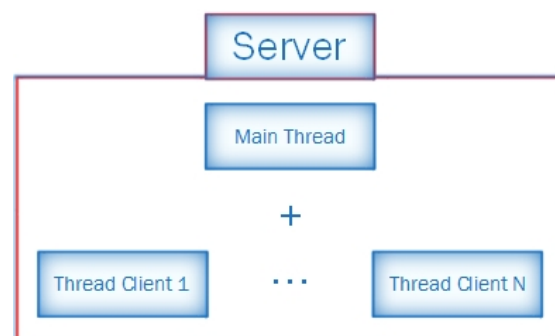


FIGURE 3.19: Server System Threads

In the figure 3.19 it is possible to see the architecture of the server. The server consists of the Main Thread, where the server will be started, and the Threads Client, which will be as many as the client number connected to the server.



- **Main Thread**

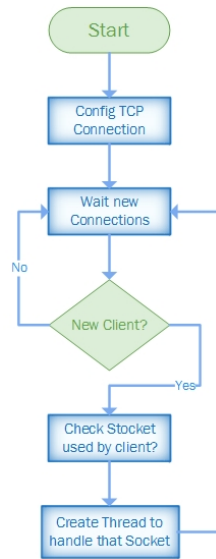


FIGURE 3.20: Server System Main Thread Flowchart

As flowchart 3.20 demonstrates, the threads that will handle the clients connected to the server will be created in the Main Thread. Its cycle is very simple, it just waits for a connection, verifies which socket the connected client used, and creates a thread to handle that client.

- **Thread Client**

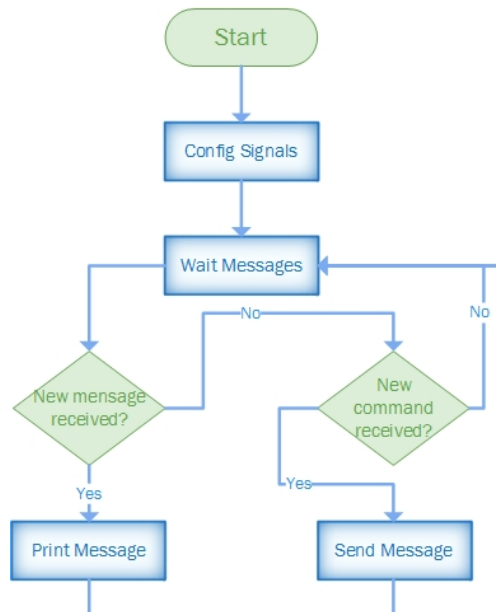


FIGURE 3.21: Server System Thread Client Flowchart

The client thread will have the same principle function of a gateway, as it is possible to see in flowchart 3.21. Its function will be to transmit messages, in this case, between the local system, and the user. Its principle of operation is, if it receives a message from the client, present it. If it receives any instruction from the user, transmit it to the local system.

### 3.7.3 GUI Application

In order to visualize what is happening on the local system, for monitoring purpose, or an error visual presenting, a GUI must be implemented. This GUI should have an interface, intuitive and easy to use, requiring no training for one to use. Like all systems described until now, it is also necessary to define its architecture, and structure of algorithms that it will use, using flowcharts.

#### Graphical Overview

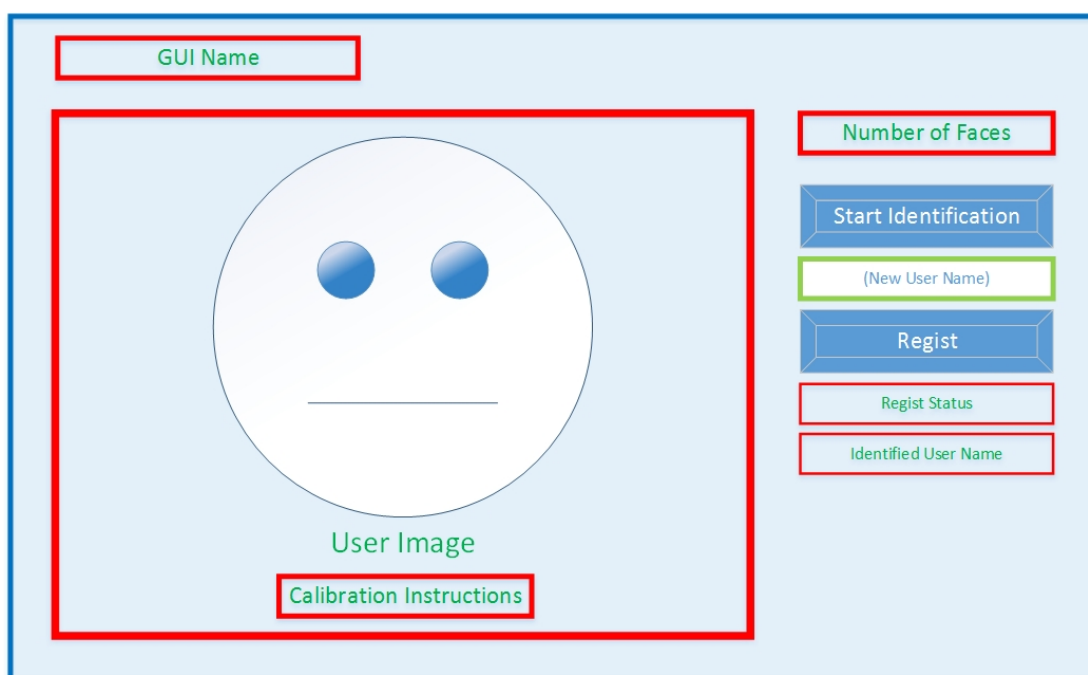


FIGURE 3.22: Graphical GUI Mockup

In the figure 3.22 it is possible to see the expected final design for the GUI. It will contain the image that the camera is capturing, the number of faces that are being identified in the acquired image and a button that will start recognition. Instructions will be displayed to the driver to put his face in the right position, with the name of the driver appearing if recognized.

The GUI will also allow the driver to be registered. By clicking on the 'Regist' button, the process will take place, first the name entered in the text box '(New Name User)' will be used to register. If successful, a message will appear, and if something has prevented the registration, will appear to.

## GUI System Architecture

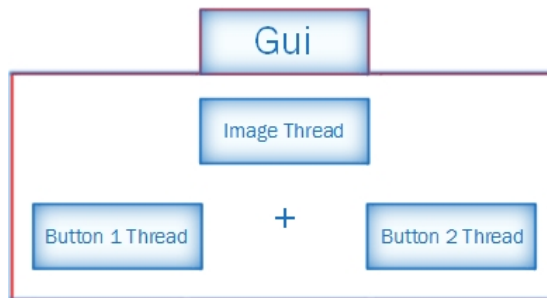


FIGURE 3.23: GUI System Architecture

As in the figure 3.23, the GUI system architecture consists on three threads. The Image Thread, which is also the Main Thread, and more. Two more threads will compose the system, the Button One Thread, which corresponds to the 'Start Identification' button and the Button Two Thread, which corresponds to the 'Regist' button.

- **Image Thread**

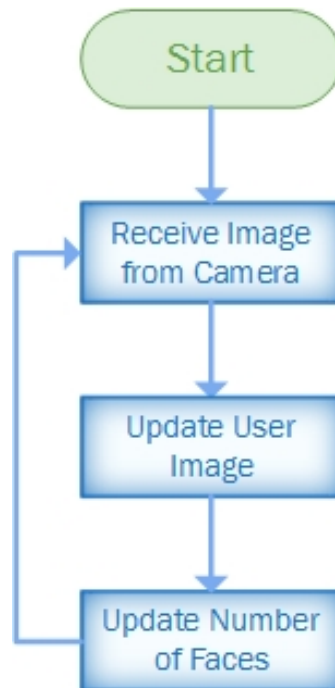


FIGURE 3.24: Image Thread Flowchart

The working principle of this thread is easy to understand (see flowchart 3.24). The image will be acquired, updating both the user image as well as the number of faces in the acquired image. Also this thread will be responsible for updating all labels in the GUI Graphical part.

- **Button One Thread**

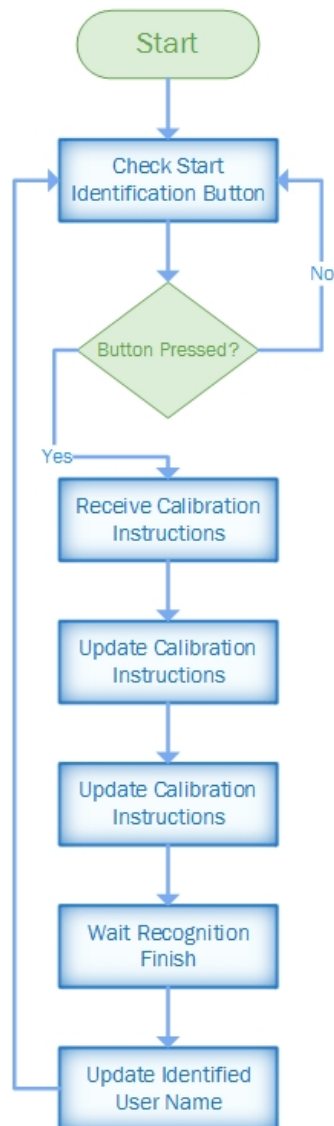


FIGURE 3.25: Start Identification Button Thread Flowchart

The Button One Thread is only responsible for identifying the 'Start Identification' button. The identification of a person will start everything. After being pressed, two labels will begin presenting information, firstly the calibration instructions to the face position, and after the recognition, the label that will present the name, of the user (flowchart 3.25).

- **Button Two Thread**

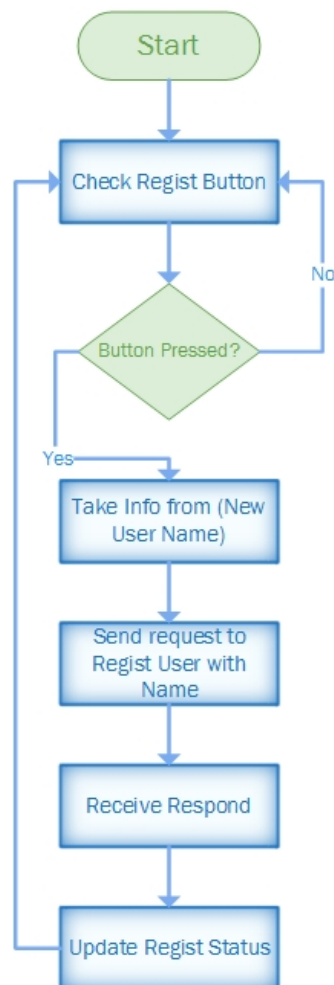


FIGURE 3.26: Regist Button Thread Flowchart

Like the previous thread, this one is responsible for identifying the change of a button, but in this case the button 'Regist'. When pressed, the text written in the text box '(New Name User)' will be used as the name to save the calculated measurements. The registration status will be displayed just below the 'Regist' button (flowchart 3.26), like it is shown in figure 3.22.

This button should only be active after an attempt of recognizing the driver is made, where in case the driver was not recognized, and deactivated after a successful registration to avoid multiple saves of the same measurements..

## 3.8 Validation Specification

In this section, it is specified the process of verification and validation of the final system. It is one of the most important phases because it will verify if the objectives of the system have been accomplished and the requirements have been fulfilled.

According to Magneti Marelli Powertrain [66], manufacture of driver control for combustion engines and transmission control systems, in the automotive area, all activities performed during validation have the same objective and are carried out based on the pre-requisites of the system. These objectives will verify:

- System functionality;
- Compliance with customer requests in terms of functionality and reliability;
- System robustness (ability of the system to tolerate the presence of failures);
- System security.

Each type of system has its distinct characteristics, therefore, the system validation activities consist of differentiated and specific activities to test the system developed. For the system of this dissertation, common tests can be realized, that are necessary to verify the content and the functions found, including:

- System communication test;
- Feature tests;
- Correct use tests;
- Diagnostic tests.

In addition to the advantages of performing tests on the developed system, validation offers advantages to those who have developed the system, in the sense that information is extracted that can be vital for the team. In the end, in addition to ensuring the integrity and quality of the developed system, validation allows:

- Extracting all available resources in the system;
- Making the team have all technical knowledge about the system;
- Documenting the technical findings;
- Exploring all possible automated resources in order to avoid the manual production step, if the system is intended for industrial production;
- Directing to action analysis necessary to a contingency plan in case of a data loss.

### 3.8.1 Validation Criteria

For the validation of the recognition system, a few criteria should be taken in consideration:

- Accuracy - value that defines the probability of the distinction of a true positive case and a true negative case during system tests:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

TP-True positives|TN-True negatives|FP-False positives|FN-False negatives

- Sensitivity - value that defines the probability of identifying a true positive case between positive cases during system tests:

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

- Specificity - value that defines the probability of identifying a true negative positive case between negative cases, during system tests:

$$\text{Specificity} = \frac{TN}{TN+FP}$$

### 3.8.2 Tests Specification

For the validation and verification of the final system, specific tests will be performed and metric values will be obtained. The tests will be performed in controlled environments in laboratory, to maintain conditions such as luminosity, temperature, distractions and numbers of users, controlled.

Using this approach makes experimental control easier, as well as recording data for the calculation of metric values, increasing the safety and efficiency of the system. This approach eliminates the real conditions in driving a real vehicle, but due to time constraints and access limitations, the priority is implementing the system.

In order to carry out the tests, users volunteering will be necessary, as well as the control during the tests by a supervisor, for the assistance and to answer any doubts that appear. Some laboratory conditions will be controlled such as, keeping the user alone only with the presence of the supervisor, no external noise to avoid distractions and control of luminousness.

Keeping all the conditions under control and obtaining the data of known people and with previous training in the system, makes the data reliable and safe for the accomplishment of the metric calculations and evaluation of the system.

### 3.8.3 Test Method

The tests performed on the system will be divided into three parts: Elements tests, Communication tests and Metrics tests.

#### Elements tests

This part aims at the testing and integration of all elements of the system, such as the camera, CPU, database and code development tool.

For the camera, it will be done a survey of all its features, ie, register other functions that the camera can have in addition to facial recognition. For the CPU, it will be done an analysis on which programs for software development it supports, and how it can make the boot of the OS faster. For the database, it will be made a survey of the structure of the messages needed to perform the data save or survey of the same. In conclusion form:

- Elements functionalities survey;
- Software tools compatibility;
- Interface between elements and message structure.

### **Communication tests**

This testing phase will focus on issues of integrating the facial recognition system with the server, which is a replacement for the cluster/central stack in the DSM (see section 3.3).

Like the previous tests, these ones will also be carried out in a laboratory environment, and will have as the objective:

- Communication tests between facial recognition system and the server, namely performing a survey of the messages sent between them, and if the protocol described in section 3.7.2 is verified;
- Activation of the facial recognition system by the server. This means that although the communication protocol is verified, the activation of the system is carried out remotely.

### **Metrics tests**

In this series of tests, the objective is to obtain information for the calculation of metric values: accuracy, sensitivity, specificity. Here, several tests will be performed, with several people, and the number of measures performed on the face will be varied, and the number of compatibilities among the users, that use the system, will be evaluated to check if there is any probability of obtaining some wrong identifications.

At the end of these tests, all the constrains of the final system will be described, and the possible actions that could improve driver identification, as well as making the facial recognition process more practical.



# Chapter 4

## System Implementation

In the fourth chapter, it is possible to find descriptions about what was implemented to achieve the final system proposed in this dissertation. Some details will be presented as well as the reason for them.

This chapter will describe the implementation of the systems that make up the final system, the code used for its implementation, description of communication protocols and description of threads.

The focus in this chapter will be given to the developed software and not to the hardware used, since this part was well described in the previous section. Only one image of the final setup will be presented to show how the laboratory tests were performed, and how the final system was developed.

### 4.1 Final Prototype System

In order to perform hardware tests on the chosen ones, before beginning the development of a final algorithm, to achieve the final system, a configuration system has to be built to acquire the information needed. In this case, before starting any final development of software, some tests were made to the camera to get its real capabilities, such as range, image quality, number of frames per second (fps), among others. For this, a test setup was created, and this setup ended up as the final setup.

The final system is simple, since it consists of the three systems referred in section 3.3: the local system, the cluster/central stack, and a GUI.

#### 4.1.1 Tests Configuration

For the confirmation of the range, using the first version of local system, the distance that the camera was capable of identifying someone's face was verified. For the FPS number, a software algorithm was used to see how many images it was capable of acquiring from the camera in a second. For the image quality, the possible number of resolutions that the camera image can acquire in visible light was verified.

To answer the question about other features that the IntelSense SR300 has, several applications from Intel were used, and after that, the results were verified.

#### 4.1.2 Materials for the Prototype System

In order to perform the laboratory tests, three components were used: a computer running the server, a monitor to present the GUI and another computer, installed with the camera, performing the local system function (see figure 4.1).

This setup was not only used in a laboratory environment, but also became the final system, that is, due to time problems, the component referred to in section 3.5.3, the Intel Mini-PC NUC, could not be acquired and was thus replaced by a computer with similar characteristics. Because the characteristics of this PC are so similar, tests on the time needed to boot the OS and run the algorithm of local system were conducted.

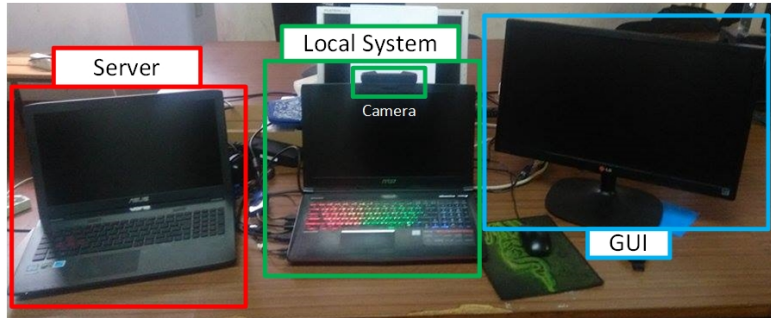


FIGURE 4.1: Laboratory Setup

## 4.2 System Architecture

In chapter 3, all the hardware, software development tools and design for graphic environment were defined, making it possible the creation of the final diagram of the system, as seen in the following figure.

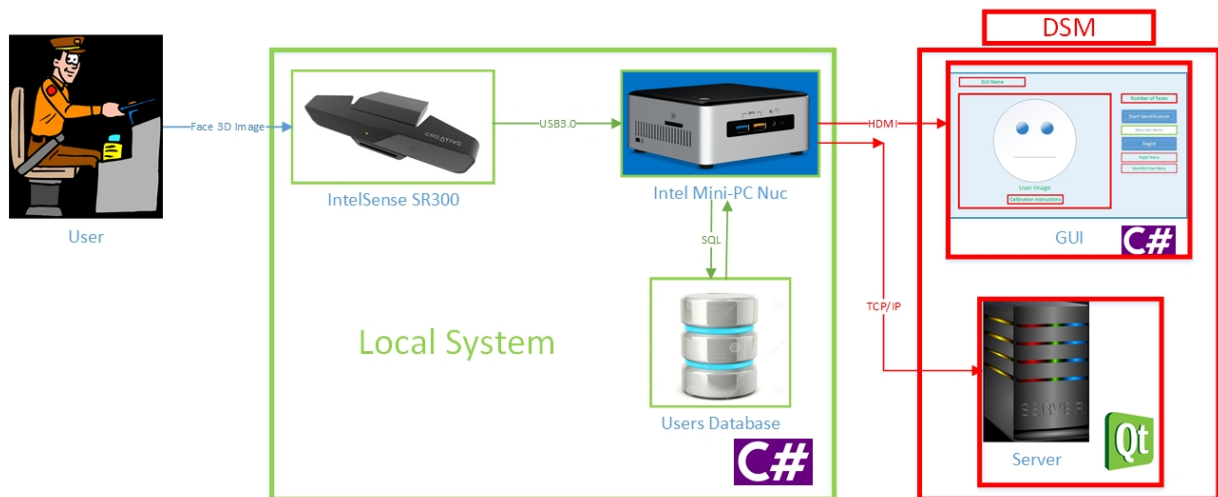


FIGURE 4.2: Final System Diagram

In the figure 4.2, it is possible to see the entire final structure, with the necessary simulators of what would be in a real case, and the methods used to communicate and display information. Comparing this figure with 3.9, it is possible to see that the cluster/central stack has been replaced by the server, the PC monitor has been replaced by a more detailed version of the information, and it is also possible to see the language, and development tool used to implement the necessary algorithms in each of the systems.

Because this image is a more explanatory version of its previous version (see section 3.7) the structure to present the implementation made in each of the systems will follow the same order, thus dividing the algorithms that will be presented in "Local System", "Server System", and "GUI Application".

## 4.2.1 Local System

In the following sections, it will be discussed the functionalities that complement the algorithm used to implement the functions expected by the local system. To start, an approach to the "Main Routine" of the system is made, and after some parts of the algorithm is made as a complement.

### Main Routine

```
static void main()
{
    ...
    ConfigureRealSense();

    //Define Timer configuration
    aTimer = new System.Timers.Timer();
    aTimer.Elapsed += new ElapsedEventHandler(SearchMeasures);
    aTimer.Interval = 1000;
    aTimer.AutoReset = false;
    aTimer.Enabled = false;

    //number of faces detected
    Int32 numFacesDetected = 0;
    Int32 currnumFacesDetected = 0;

    //index of the face
    Int32 index_face = 0;

    ...

    //Prevent spamming the server with calibration messages
    String lblWarnings_content_last = lblWarnings_content;

    // Start AcquireFrame/ReleaseFrame loop
    while (senseManager.AcquireFrame(true) >= pxcmStatus.PXCM_STATUS_NO_ERROR)
    {
        ...
    }
}
```

LISTING 4.1: Main Routine Brief

The above extract is from the main routine and it is possible to see, after the initialization of some variables that will be used throughout the function, that the camera settings will be defined, in the function "ConfigureRealSense()".

After these settings, a timer called "aTimer" is defined. This timer will be used later to set a stop in the calculation of the measurements of the driver's face. In the instruction "aTimer.Elapsed += new ElapsedEventHandler (SearchMeasures)", the timeout of "aTimer" is associated with function "SearchMeasures" , ie, when the timer "aTimer" reaches one second ("aTimer.Interval = 1000" defines a count of 1000 millisecond = 1 second), the "SearchMeasures" function is executed. The remaining two instructions associated with "aTimer" serve to prevent a new count after the timeout, and set it to its initial state, off.

In this function other variables will be initialized, some to be equal to the number of faces identified in the image acquired by the camera, others to navigate the structure that keeps each of the faces saved. These variables will be explained later.

After all the initializations, the local system will enter in a loop, which in a brief explanation, consists in receiving instructions from the server, acquiring images from the camera, processing those images and then sending this information to the server.

## Set Configuration

```
private void ConfigureRealSense()
{
1   PXCMFaceModule faceModule;
2   PXCMFaceConfiguration faceConfig;

   // Start the SenseManager and session
   senseManager = PXCMSenseManager.CreateInstance();

   // Enable the color stream
4  senseManager.EnableStream(PXCMCapture.StreamType.STREAM_TYPE_COLOR, 640, 480, 60);
   //STREAM_TYPE_DEPTH/STREAM_TYPE_IR/STREAM_TYPE_COLOR

   // Enable the face module
   senseManager.EnableFace();
   faceModule = senseManager.QueryFace();
   faceConfig = faceModule.CreateActiveConfiguration();

   // Configure for 3D face tracking
8  faceConfig.SetTrackingMode(PXCMFaceConfiguration.TrackingModeType.FACE_MODE_COLOR_PLUS_DEPTH);
   faceConfig.landmarks.IsEnabled = true;
   faceConfig.QueryPulse().Enable();
   // Enable facial recognition
   recognitionConfig = faceConfig.QueryRecognition();
   recognitionConfig.Enable();

   // Apply changes and initialize
   faceConfig.ApplyChanges();

   //inicialize database
14  dbConnection = new SQLiteConnection("Data Source=C:/Users/Pedro/Desktop/FaceID_0_3/FaceID.db");
15  dbConnection.Open();

16  if (dbConnection == null || dbConnection.State == System.Data.ConnectionState.Closed) {
       lblNumFacesDetected.Content = String.Format("Database not available");
       return;
   }

19  try{
       client = new TcpClient(hostName, portNum);
       nwStream = client.GetStream();
       client_connection_status = 1;
   }
   catch (Exception e) {
       lblNumFacesDetected.Content = String.Format("Server TCP Down");
   }

25  if (senseManager.Init() < 0) {
       lblNumFacesDetected.Content = String.Format("No camera available");
       return;
   }
   faceData = faceModule.CreateOutput();

   // Mirror image
29  senseManager.QueryCaptureManager().QueryDevice().SetMirrorMode
(PXCMCapture.Device.MirrorMode.MIRROR_MODE_HORIZONTAL);

   // Release resources
```

```

    faceConfig.Dispose();
    faceModule.Dispose();
}

```

LISTING 4.2: Camera/Database/Server Connection Configuration

Analyzing the code presented, it is possible to verify that it represents the function for camera, database and connection with the server configurations. For the analysis of this function, a three-stage explanation will be given: firstly to the camera instructions, followed by the instruction related to the database, and finally to the TCP connection.

### • IntelSense SR300 Camera

The first instructions made in the function "ConfigureRealSense" are the creation of the objects, "faceModule" and "faceConfig". The first variable allows access to Intelisense library functions (section 3.6.4) to perform the tracking and detection of faces and the second variable allows the access to functions for definition and change of the tracking module.

In the fourth instruction, the definition of the type of image that will be presented in the GUI is done, thus choosing the "STREAM\_TYPE\_COLOR", which represents the image in visible light. In this instruction, the resolution of the image, "640x480", and the number of frames of the image, "60" are set. In the table 4.1, it is possible to see the resolutions, FPS and image types that can be taken from the camera.

Stream Type Image	Image Resolution	FPS
STREAM_TYPE_COLOR (Visible Image)	Maximum 1920 x 1080 1080p	30 to 60 (in maximum resolution only 30)
STREAM_TYPE_DEPTH (Depth Image)		
STREAM_TYPE_IR (Infrared Image)		
STREAM_TYPE_LEFT (Infrared Image from Left Camera)		
STREAM_TYPE_RIGHT (Infrared Image from Right Camera)		

TABLE 4.1: Stream Types to SR300 Intel camera

As can be seen in the table 4.1, it is possible to display five different streams of the camera. To allow the streaming of more than one image type, simply make the instruction 4 again.

One more thing that can be seen through the table, is the ability to stream an image from the left camera, or from the right camera, ie, the SR300 has two infrared cameras. The reason for them to contain these two cameras is to capture the 3D image of what is in front of them, using a technique already described in the section 2.2.3.

In instruction eight, it is defined the type of image that will be used to make the face tracking, which in this case, for the tracking it will be used the visible image along with depth information, that is, a 3D image. Two more types of inputs can be used for facial tracking: only the visible light image (FACE\_MODE\_COLOR) or a static image/picture/image file (FACE\_MODE\_COLOR\_STILL).

In instruction 25, it is verified not only whether the camera has the capability to perform face tracking, but if it is also available and ready for use by the local system. If it is not, the information will be passed to the GUI to be presented.

In order to not confuse a user who is viewing the stream camera image in the GUI, it was decided make the acquired image mirror, this way the image has the same effect as when a person looks in the mirror.

### • Database

In the function "ConfigureRealSense" it is defined the path to the database that the local system will use to save the users that it recognizes, instruction 14, after it has performed the instruction 15 that makes/tries the connection to the database, and instruction 16 is used to check whether this connection was successful. If it has not been successful, local system displays in the GUI the information that was not able to connect.

To indicate the path to the database, it is necessary to create a database in a folder. In this case the SQLite database was used. It is a small file with the data storage structure of a database. This database does not need a username or password to connect to it, only needing the connection. For this project, a structure had to be created in the database to store in an orderly way the measurements taken from the driver, thus a command was made to create a table in the database with the desired structure.

```
CREATE TABLE UsersCar
(
UserName TEXT NOT NULL ,
Measure1 blob NOT NULL ,
Measure2 blob NOT NULL ,
Measure3 blob NOT NULL ,
Measure4 blob NOT NULL ,
Measure5 blob NOT NULL ,
Measure6 blob NOT NULL ,
Measure7 blob NOT NULL ,
Measure8 blob NOT NULL ,
Measure9 blob NOT NULL ,
Measure10 blob NOT NULL ,
Measure11 blob NOT NULL ,
Measure12 blob NOT NULL ,
Measure13 blob NOT NULL ,
Measure14 blob NOT NULL ,
Measure15 blob NOT NULL ,
Measure16 blob NOT NULL ,
Measure17 blob NOT NULL ,
Measure18 blob NOT NULL ,
Measure19 blob NOT NULL ,
Measure20 blob NOT NULL ,
PRIMARY KEY (UserName)
)
```

LISTING 4.3: SQL Command to create a table

As it is possible to analyze by the extract of code above, the table was created with a first field that will indicate the user's names and twenty remaining fields that will indicate their measurements. A table row will represent a single user, and as it is possible to see in the code extract, the "UserName" field is the primary key, that is, it is impossible to insert two rows in the table with the same "UserName" field, thus preventing the registration of different users with the same name.

For a more graphical view of the contents of the table, the DB Browser program described in the section 3.6.3 was used.

### • TCP Connection

For the connection to a TCP server, it is necessary to know two types of information about the proper server. Firstly, where IP address is hosted, and then the port dedicated to the TCP connection.

In instruction 19, an attempt is made to connect to the server. The IP of the server is given by the variable "hostName" and the port by the variable "portNum". If the connection is not performed, this information will once again be passed to the GUI and displayed. If the connection is made, a variable will change its value to 1. This variable, "client\_connection\_status" will provide the TCP connection status information in many situations, in order to decide, whether or not to send messages to the server.

client_connection_status (value)	Status
0	No connection
1	Connection made
2	Recognition requested
3	Recognition in progress
4	User name requested

TABLE 4.2: TCP connection progress regist

As we can see in the table 4.2, the variable "client\_connection\_status" has the value 0 when no connection was made to the server and 1 when the connection was successful. If the local system received a message from the server requesting a driver acknowledgment the variable takes the value 2, and 3 when the recognition process is in progress. It will have the value 4 when, before a recognition try, the local system concludes that it is a new driver and requests the server a name to associate with the measures taken/calculated.

### IDLE Process

```

...
1  faceData.Update();
2  numFacesDetected = faceData.QueryNumberOfDetectedFaces();

3  if (numFacesDetected > 0) {
4      if (currnumFacesDetected == 0 || currnumFacesDetected != numFacesDetected) {
            index_face = identify_face(faceData, numFacesDetected);
            currnumFacesDetected = numFacesDetected;
        }

        // Get the closest face detected
7      PXCMMFaceData.Face face = faceData.QueryFaceByIndex(index_face);

...

```

LISTING 4.4: Main Loop Brief

The code extract above represents a small part of the loop of instructions that the local system performs. As already mentioned in the section 4.2.1, the "faceData" instance contains functions for tracking and detection of faces in an image, so after the acquisition of a new image, the instruction "faceData.Update()" is made. This instruction will

analyze the new image and record the faces it contains. In instruction 2, the number of faces found in the new image is registered on the variable "numFacesDetected".

When an image acquired by the camera has more than one face, there is a problem. In the case of an recognition request, the face meant to be recognized is not defined. For this it was created another variable that also contains number of faces, the variable "currnumFacesDetected". This variable will only save the number of faces found in the last image acquired.

In instruction 4, the problem of multiple faces is solved by a simple comparison. If the number of faces identified in the last image is different from the number identified in the current image, the function "identify\_face" will be executed, which will return the face closer to the camera.

The example considered for the implementation of this resolution was: if the driver enters the car with his/her children, the local system will try to know which face is closest to the camera and assumes that it is the driver. If the number of faces changes, the system assumes that the driver may have left and looks for the next closest face again.

The function "identify\_face" receives as a parameter a structure with all the faces found and after that, the execution returns the index of the closest face in this structure.

In instruction 7, based on the return of the "identify\_face" function, the face closest to the camera is obtained.

## Multi Face

```

public int identify_face(PXCMFaceData faceData_aux, int numberoffaces)
{
    //z coordinate of the most close landmark point
    double z_most_close = 100;
    //index of face to detect
    int index_face = 0;

    //more then 1 face identified
    for (int i = 0; i < numberoffaces; ++i) {
        // Get the first face detected (index i)
4         PXCMFaceData.Face first_face = faceData_aux.QueryFaceByIndex(i);

        // Retrieve the landmarks from the front face
5         PXCMFaceData.LandmarksData faceLandmarkData = first_face.QueryLandmarks();

        if (faceLandmarkData != null) {
            // Get the points
            PXCMFaceData.LandmarkPoint[] points;
            faceLandmarkData.QueryPoints(out points);

            if (points[29].world.z < z_most_close) {
                z_most_close = points[29].world.z;
                index_face = i;
            }
        }
    }
    return index_face;
}

```

LISTING 4.5: Identifying the closest face

As stated in the section 4.2.1, the function presented in the above code extract gives answers in knowing which face is closest to the camera, in case of several people in the image being acquired and the need to perform driver recognition arises.



For this function two things are needed, the structure with all the faces identified in the new acquired image, "PXCMFaceData faceData\_aux", and the number of faces identified, "int numberoffaces". In addition, some auxiliary functions that can be found in the Intellisense library are needed.

In the beginning, some inicializations are made on some auxiliary variables required by the process, the variables "z\_most\_close" and "index\_face", and after that a cycle begins. In this cycle, one face of the structure (intruction 5) is collected and, using the function "QueryLandmarks()" from the Intellisense library, the face's landmarks are obtained. After that, a comparison is made between the Z coordinate of lankmark point 29 and "z\_most\_close", and if it has a smaller value, "z\_most\_close" takes its value and the index of the face is saved in "index\_face", otherwise it is discarded.

When the function "QueryLandmarks()" is executed, a set of points about the identified face are created. The information of each of these points is saved in the array "PXCMFaceData.LandmarkPoint[] points" and for each of the positions of the array has:

```
struct LandmarkPoint {
    LandmarkPointSource source;
    pxcI32 confidenceImage;
    pxcI32 confidenceWorld;
    PXCPint3DF32 world;
    PXCPintF32 image;
};
```

LISTING 4.6: Landmark point struct

In the variable "source" it describes the name of the landmark, "world" the 3D coordinates of the point (in meters) and "image" the coordinates of the point in the acquired image (in pixels). Through previous tests to the camera it was noted that point 29 indicated the nose, making it the closest point of the camera when the face is directed towards it. Also, using a reference of the image given by Intel 4.3, it is possible to verify that the Z coordinate represents the distance between the object being analyzed and the camera, so it was concluded that the value "points[29].world.z" represents the distance from the nose of the face under analysis to the camera, meaning that the face with the lowest value at that point, was the closest.

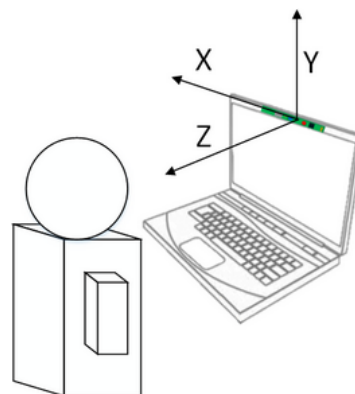


FIGURE 4.3: Camera vision axes [29]

## Face Position

With the necessary configuration made to the camera, and with the indication of which face will be recognized,, the local system is ready to achieve its main objective.

As already described in the section 3.7.1, when a request for driver recognition is received, three phases occur: corrections on the face position to be recognized, calculation of the face measurements and lastly, database search.. This same division will be applied to explain the algorithm implemented.

```

...

float headRoll = 10;
float headPitch = 10;
float headYaw = 10;

// Retrieve pose estimation data
4  PXCMFaceData.PoseData facePoseData = face.QueryPose();

if (facePoseData != null) {
    PXCMFaceData.PoseEulerAngles headAngles;
    facePoseData.QueryPoseAngles(out headAngles);
    headRoll = headAngles.roll;
    headPitch = headAngles.pitch;
    headYaw = headAngles.yaw;
}

11 if (headRoll < 3 && headRoll > -3 && headPitch < 3 &&
headPitch > -3 && headYaw < 3 && headYaw > -3) {

    lblWarnings_Visibility = true;
    lblWarnings_content = "Scan Available";
    lblWarnings_color = "#FF00FF00";

}
else {
    lblWarnings_Visibility = true;
    lblWarnings_color = "#FFFF0000";

    if (headRoll > 3) lblWarnings_content = "Rotate right";
    if (headRoll < -3) lblWarnings_content = "Rotate left";
    if (headPitch > 3) lblWarnings_content = "Look down";
    if (headPitch < -3) lblWarnings_content = "Look up";
    if (headYaw > 3) lblWarnings_content = "Turn right";
    if (headYaw < -3) lblWarnings_content = "Turn left";
}

if( points[29].world.z > 0.5 || points[29].world.z < 0.4) {
    lblWarnings_Visibility = true;
    lblWarnings_color = "#FFFF0000";

    if (points[29].world.z > 0.5) lblWarnings_content = "Move closer";
    if (points[29].world.z < 0.4) lblWarnings_content = "Too much closer";
}

...

```

LISTING 4.7: Calibration/Position head warnings

To correct the head position of the person meant to be identified, a function of the library Intellisense is used. The function "face.QueryPose()" is used on the closest face and the function returns three datas: Roll, Pitch and Yaw (instruction 4 of the code shown above). It is possible to see in the figure 4.4 what each data is.

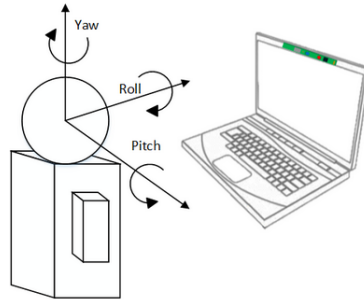


FIGURE 4.4: Head pose [29]

These three data are used to give the user/driver indications on what he has to do to align with the camera. These values come in degrees and, as it is possible to see in instruction 11, it is intended for the user to be aligned between -3 and +3 degrees in every way, setting a small margin of maneuver for the user, but not fixed, since it is hard for someone to keep their head static for some time.

If the user brakes any of these impositions, he will receive a warning in the GUI or through the server.

Another imposition is the distance between the driver and the camera, since when the camera attempts to recognize with the user very close, it can lead to the face not being identified, or when at long distances, it can lead to a wrong recognition of the driver. For this, the technique already described in the previous section, was used again. Once again the distance from the user's nose to the camera, is calculated, and if it is between 40 and 50 cm, no warning will be sent, otherwise this warning will be the higher priority message.

## Face measurements

```

...
measure_process = true;
aTimer.Enabled = true;

for (int i = 0; i < size_measures; ++i) {
    currmeasure[i] = 0;
}

frameNum = 0;

////////////////////////////////////

// Get the points
PXCMFaceData.LandmarkPoint[] points;
faceLandmarkData.QueryPoints(out points);
....

if (measure_process == true) {
9     measure[0] = CalculateDistance(points[4].world, points[14].world, 3);
    measure[1] = CalculateDistance(points[9].world, points[2].world, 3);
    measure[2] = CalculateDistance(points[4].world, points[26].world, 3);
    measure[3] = CalculateDistance(points[9].world, points[26].world, 3);
    measure[4] = CalculateDistance(points[4].world, points[10].world, 3);
    measure[5] = CalculateDistance(points[9].world, points[18].world, 3);
    measure[6] = CalculateDistance(points[4].world, points[32].world, 3);
    measure[7] = CalculateDistance(points[9].world, points[33].world, 3);
    measure[8] = CalculateDistance(points[10].world, points[18].world, 3);
    measure[9] = CalculateDistance(points[5].world, points[10].world, 3);

```

```

measure[10] = CalculateDistance(points[0].world, points[18].world, 3);
measure[11] = CalculateDistance(points[26].world, points[31].world, 3);
measure[12] = CalculateDistance(points[30].world, points[32].world, 3);
measure[13] = CalculateDistance(points[14].world, points[30].world, 3);
measure[14] = CalculateDistance(points[22].world, points[32].world, 3);
measure[15] = CalculateDistance(points[4].world, points[0].world, 3);
measure[16] = CalculateDistance(points[5].world, points[9].world, 3);
measure[17] = CalculateDistance(points[33].world, points[61].world, 3);
measure[18] = CalculateDistance(points[39].world, points[61].world, 3);
measure[19] = CalculateDistance(points[33].world, points[39].world, 3);

for (int i = 0; i < size_measures; ++i) {
    currmeasure[i] = (currmeasure[i] * frameNum + measure[i]) / (frameNum + 1);
}

//increase number of frames
frameNum++;
}
...

```

LISTING 4.8: Calibration/Position head warnings

With the user's face in the correct position, the calculations of the driver's face measurements start. To begin this procedure some essential variables must be started. The variable that counts the number of frames collected from the beginning of the recognition process, "frameNum", is equalized to zero, as well the array of doubles that will contain the average of each measure, "currmeasure". After the initializations, the timer previously mentioned in the section 4.2.1, "aTimer", starts counting, and the variable that allows calculate the measures, is set to "true".

The calculations made on the users face are done based on their landmarks, and as already mentioned in section 3.7.1, the coordinates of these points are sometimes not very stable. Although there have variations in the order of millimeters, when it comes to performing measurements on a person's face, these are considerable variations. To mitigate this problem, an average of calculated measurements is made for 1 second (timeout of timer "aTimer"). In the end the result is the array "currmeasure" with 20 averages about the user (see figure 4.5).

Tests performed in the laboratory showed an acquisition rate of approximately 51 frames in a second, that is, during that second,  $51 \times 20 = 1020$  measurements were calculated. In order to avoid an immense structure to keep these measurements saved in an orderly way, another technique was made which allows a dynamic calculation average measurements, using the equation:

$$\text{New\_Average} = \frac{\text{Old\_Average} \times \text{Total\_samples} + \text{New\_Measure}}{\text{Total\_samples} + 1}$$

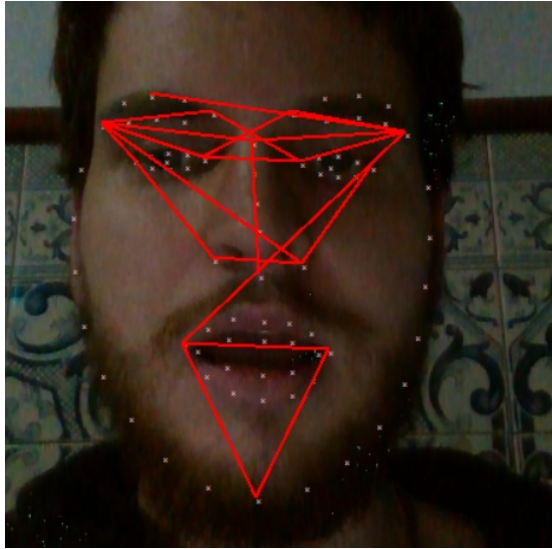


FIGURE 4.5: The 20 face measures acquired

As it is possible to see in the code above, the instructions 9 to 28 use the function "CalculateDistance" to calculate the distance between 2 points/landmarks. The function receives the 3D coordinates of two points and the number "3", that indicates that the distance calculation must be performed in 3 dimensions.

```
public double CalculateDistance(PXCMPPoint3DF32 p1, PXCMPPoint3DF32 p2, int dim)
{
    if (dim == 2)
        return Math.Sqrt(Math.Pow(p2.x - p1.x, 2.0f) + Math.Pow(p2.y - p1.y, 2.0f))*1000;
    if (dim == 3)
        return Math.Sqrt(Math.Pow(p2.x - p1.x, 2.0f) + Math.Pow(p2.y - p1.y, 2.0f) + Math.Pow(p2.z - p1.z, 2.0f))*1000;
    return 0;
}
```

LISTING 4.9: Landmark point struct

As described in the code presented above, after receiving the 3D coordinates of the two points and the indication to perform the calculation in 3 dimensions, the following formula is executed:

$$\text{Distance} = \sqrt{(X2 - X1)^2 + (Y2 - Y1)^2 + (Z2 - Z1)^2}$$

Since the results of this equation came in meters, a multiplication by 1000 is made at the end, to make the value a more substantial amount and in the millimeter order.

## Face Checking

```
public void SearchMeasures()
{
    //Console.WriteLine("\nisecond already passed .... Saving Measures ...");

    SQLiteDataReader myReader = null;
    String myQueryInsert = "";
    SQLiteCommand command = new SQLiteCommand(myQueryInsert, dbConnection);

    for (int i = 0; i < 100; i++) { ArrayUsersNames[i] = null; ArrayUsersPresence[i] = 0; }
```

```

last_index_user = 0;

for (int i = 0; i < size_measures; i++) {
7   myQueryInsert = "Select * From UsersCar Where (Measure"+ (i+1).ToString() +
    " between " + currmeasure[i] + "-2.5 and " + currmeasure[i] + "+2.5)";

    command = new SQLiteCommand(myQueryInsert, dbConnection);
    myReader = command.ExecuteReader();
10  while (myReader.Read()) {
        int index_aux = -1;
        for (int j = 0; j < last_index_user; j++) {
            if (ArrayUsersNames[j] == myReader.GetString(0)) index_aux = j;
        }
        if (index_aux == -1) {
            ArrayUsersNames[last_index_user] = myReader.GetString(0);
            last_index_user++;
        }
        else ArrayUsersPresence[index_aux]++;
    }
}

int index = -1;
19  for (int i = 0; i < last_index_user; i++) {
    if (ArrayUsersPresence[i] >= 15) {
        index = i;
        Console.WriteLine("User: {0} Presence: {1}",
            ArrayUsersNames[i], ArrayUsersPresence[i]);
        break;
    }
}

24  if (index != -1) {
    UserName = "User Identified: " + ArrayUsersNames[index];
26  myQueryInsert = "Update UsersCar Set Measure1=" + currmeasure[0];
    for (int i = 1; i < size_measures; i++) {
        myQueryInsert = myQueryInsert + " , Measure" +
            (i + 1).ToString() + "=" + currmeasure[i];
    }
    myQueryInsert = myQueryInsert + " Where UserName='" + ArrayUsersNames[index] + "'";
    command = new SQLiteCommand(myQueryInsert, dbConnection);
    if (command.ExecuteNonQuery() > 0) Console.WriteLine("Update with success");
    else Console.WriteLine("Update without success");

}
else UserName = "None User Identified";

...

Console.WriteLine("Frames used: " + frameNum);

measure_process = false;
aTimer.Enabled = false;
frameNum = 0;
}

```

LISTING 4.10: Landmark point struct

With the face in the correct position, the measurements are calculated, and finally the timeout of the "atimer" timer occurs (see section 4.2.1). With the timeout, the function "SearchMeasures" is executed and this function has as main objective of searching the measurements previously calculated, on the database.

For manipulation and execution of SQL code in the database, functions and structures of the library SQLite are used. As it is possible to see from the extract of the previous code, two instances and a variable are created. The string "myQueryInsert" is used to "write" the SQL command to insert and update data in the database. The instance "command" is used to execute those commands and the instance "myReader" is used to read the database output.

Remembering the contents of the section 3.7.1, for the search of the driver in the database, two arrays are used, "ArrayUsersNames" to store the database user's name and "ArrayUsersPresence" for the number of compatibilities of those users, with the driver.

Instead of writing a SQL command that searches for 20 identical measurements in the database, collected from the driver, a search for each one is performed. First it writes the query (command SQL) that search for users, in the database, that have the first measure identical to the first driver's measurement. To search for the information the command used is:

```
"Select * From UsersCar Where (Measure1 between " + currmeasure[0]
+ "-2.5 and " + currmeasure[0] + "+2.5);"
```

LISTING 4.11: Query to search for the first measure

The response to this command can be null, that is, no user in the database has the "Measurement 0" identical to the driver. Imagining that two users, John and Tony, have an identical match, the output in the arrays after the cycle in the instruction 10, will be:

	Index	0	1
ArrayUsersNames	String	John	Tomy
ArrayUsersPresence	Int	1	1

TABLE 4.3: Arrays Content before Measure1 check

After searching for the first measurement, the same process is performed with the second one. The command for searching for the second measurement is:

```
"Select * From UsersCar Where (Measure2 between " + currmeasure[1]
+ "-2.5 and " + currmeasure[1] + "+2.5);"
```

LISTING 4.12: Query to search for the second measure

Assuming that the response to this command is just user Tomy, the output in the arrays after the loop is:

	Index	0	1
ArrayUsersNames	String	John	Tomy
ArrayUsersPresence	Int	1	2

TABLE 4.4: Arrays Content before Measure2 check

And this process repeats itself until the number of measurements is over. At the end, the contents of the array "ArrayUsersPresence" are analyzed to look for an index where the number is higher or equal than 75 % of the total measurements made to the driver. Remembering that the measurements are in the order of millimeters, the search for a measurement in the database is performed in a range between -2.5 and 2.5 millimeters of the measurement of the driver. This range minimizes even more the problem of landmarks instability and does not affect the correct recognition of the driver.

In instruction 19, the search for an index in the array "ArrayUsersPresence", where it contains a value equal or higher than 15, is made. The reason for this number is because

15 is 75% of 20, which is the total number of measurements calculated at the time for the driver. The first index that responds to this condition is stored in the variable "index" and, in the instruction 24, its value is analyzed and the display of the name "ArrayUsersNames[index]" is performed. This name is the identity that the local system assumes to be the driver and then updates the values in the database with the driver measurements, to avoid the problems already mentioned, the measurements change with age.

The update to the 20 measurements is done simultaneously using the command:

```
"Update UsersCar Set Measure1=" + currmeasure[0] + " , Measure2 =" + currmeasure[1]
  + " , Measure3 =" + currmeasure[2] + ... + " , Measure20 =" + currmeasure[19]
  + " Where UserName='" + ArrayUsersNames[index] + "'";
```

LISTING 4.13: Query to update all measures

And the update is done, using the name of the User saved in "ArrayUsersNames[index]", as the method of identification.

After the searching for the measurement, if the driver is recognized, all the process described above happens, and if it does not, the registration process can be performed, if desired.

## Register User

```
public void Regist(String message)
{
    String UserNameInsert = textBox.Text;
    Match match = Regex.Match(UserNameInsert, "[^a-zA-Z0-9*$]", RegexOptions.IgnoreCase);
    if (UserNameInsert == null || UserNameInsert == "NONE")
        lblUserNamePossible.Content = String.Format("Choose a name");
    else if (match.Success || UserNameInsert.Contains(" "))
        lblUserNamePossible.Content =
            String.Format("Choose a name without special characters");
    else {
        String myQueryInsert = "Insert or ignore Into UsersCar Values ('"
            + UserNameInsert + "'";
        for (int i = 0; i < size_measures; i++) myQueryInsert = myQueryInsert +
            "," + currmeasure[i].ToString();
        myQueryInsert = myQueryInsert + "));";
        SQLiteCommand command = new SQLiteCommand(myQueryInsert, dbConnection);
        if (command.ExecuteNonQuery() > 0) {
            lblUserNamePossible.Content = String.Format("User registered with success ");
            ...
        }
        else lblUserNamePossible.Content = String.Format("Not Inserted, choose another name");
    }
}
```

LISTING 4.14: Landmark point struct

If the driver is not recognized, the process for its registration can be started. For the driver's registration, only a name is required for the association of the previously calculated measurements in the database. By analyzing the code extract presented above, which represents the function that registers the driver in the database, in instructions 2, 3 and 5, some restrictions are imposed to avoid insertions of names not allowed and problems with the TCP connection.



The name used to insert measurements can not contain spaces, nor special characters, such as '\_' or '-', can not be null and can not be equal to 'NONE', since this name is part of TCP indication command to the server, that the user was not recognized.

If any of the above conditions is verified a warning is sent, otherwise the following SQL command is executed:

```
"Insert or ignore Into UsersCar Values ('NAME_CHOSEN' + "," + currmeasure[0]
+ "," + currmeasure[1] + ...+ "," + currmeasure[19] + ");"
```

LISTING 4.15: Query to update all measurements

The output to this command can indicate the existence of a user with the same name, in which case a warning will be sent and the insertion in the database will not occur, or that the insertion was successfully made.

## 4.2.2 Server System

Remembering what was written in the section 3.7.2, the server consists of a "Main thread" and its "Thread Clients", which are as many as the number of TCP clients that try to connect to the server. The server is multi-thread because its goal is to simulate DSM, which also has implemented on its system a multi-thread client.

In the following sections, it will be presented the code of each thread and an explanation of "why", and "how" it was implemented.

### Main routine

```
void MyServer::startServer()
{
1   int port = 5000;

2   if(!this->listen(QHostAddress::Any,port)) {
       qDebug() << "Could not start server";
   }
   else {
       qDebug() << "Listening to port " << port;
       qDebug() << "Waiting for connection...";
   }
}

void MyServer::incomingConnection(qintptr socketDescriptor)
{
   // new connection
   qDebug() << "Connecting";

3   MyThread *thread = new MyThread(socketDescriptor, this);

   // connect signal/slot
   // once a thread is not needed, it will be deleted later
4   connect(thread, SIGNAL(finished()), thread, SLOT(deleteLater()));

5   thread->start();
}
```

LISTING 4.16: Server Main routine

In the code extract presented above, it is possible to see how the implementation of the server, that intends to simulate DSM, is performed.

In first place, it is defined the port reserved for the TCP connection of the server, which is a 16-bit positive integer, capable of varying from 1 to 65535, but it can not be

0 because that port is already reserved. This port definition is performed in instruction 1.

In instruction 2, the server is started on the defined port, on the IP where the code is running ("QHostAddress :: Any"). If it starts successfully, a message will be presented, where it is possible to see the port reserved and that is already available to receive connections.

If a connection is received, the "incomingConnection" function is executed. In this function, a thread is created to perform the handling of the socket used by the client, instruction 3. Even before the execution of the function stops, the signal that the thread has finishes the execution is connected to the function that erases it, that is, when the thread finish objective for which it was assigned, it is immediately deleted (instruction 4).

With the signal connected and the socket set, the thread is started (instruction 5).

## Client routine

```

void MyThread::run()
{
    // thread starts here
    qDebug() << "...";

    socket = new QTcpSocket();

    // set the ID
1  if(!socket->setSocketDescriptor(this->socketDescriptor)) {
        // something's wrong, just emit a signal
        emit error(socket->error());
        return;
    }

    // connect socket and signal
    //This makes the slot to be invoked immediately, when the signal is emitted.

2  connect(&notifier, SIGNAL(activated(int)), this, SLOT(text()));
3  connect(socket, SIGNAL(readyRead()), this, SLOT(readyRead()), Qt::DirectConnection);
4  connect(socket, SIGNAL(disconnected()), this, SLOT(disconnected()));

    // I will have multiple clients, I want to know which is which
    qDebug() << "Client is connected";
5

    // make this thread a loop,
    // thread will stay alive so that signal/slot to function properly
    // not dropped out in the middle when thread dies

    exec();
}

void MyThread::text()
{
6  QTextStream qin(stdin);
7  QString line = qin.readLine();
  QByteArray auxLine = line.toLatin1();
  qDebug() <<"Data sent from server: " << auxLine;
8  socket-> write(auxLine + "\n");
}

```

LISTING 4.17: Client Thread routine

By imagining the case where a client has connected to the server, a thread is created and its behavior can be analyzed in the extract of code presented above.

The first action of the thread is to check if it received a valid socket from the server, then it carries out instruction 1, where it tries to define its socket equal to what the server

sent to it. If an error occurs with the definition, the thread immediately terminates its execution.

After the definition, some signals of the socket are associated with functions of the thread, ie, in the case of instruction 4, if the client sends some message to the server, the request activates the function "readyRead", and in this function the message will be read and displayed in a console

Other signals are connected, such as the indication that the client has disconnected (instruction 5) and that something has been written in the console data .

The signal for writing in the console data is extremely important because without it, it would be very difficult to perform tests between the local system and the server. As can be see in instruction 2, the signal is connected to the function "text". In this function, a variable is defined that will contain what was written in the console data (instruction 6) and after that, the information, in instruction 7, is read to a String, and after some conversions it is sent to the client.

In the console it will appear two types of information: prints of the messages sent in the TCP connection, and what was written in the console data that the server sent to client.

## Communication Protocol

The communication protocol that was used on the local system and on the server has been defined in the section 3.7.2. But in that section it was not specified the structure of each command for each case. In the table 4.5, it is possible to see the commands that the local system sends to the server. These commands are sent in several phases of the local system algorithm. The first command starts the execution, the next eight are the commands sent during the "Face Position" process and the next one when the measurements are to be performed. If the local system receives a request for recognition, but does not find any face in the acquired image, the command "#FACE\_NULL" will be sent.

During the registration process of the driver, 3 commands can be sent, meaning that registration was made with success, and that the driver must choose another name because what he chose is already in use. The reason for the existence of two commands to inform the registration denial is because it is necessary to differentiate when the name was not registered because the user's name already exists on system, or the given name has strange characters or denominations.

TCP Command	Fuction
#FACE_RECOG_ON	Local system ready to receive request
#FACE_MOV_CLOSER	Driver move closer
#FACE_MOV_FAR	Driver to much closer
#FACE_ROT_RIGHT	Driver rotate head right
#FACE_ROT_LEFT	Driver rotate head left
#FACE_TURN_DOWN	Driver look down
#FACE_TURN_UP	Driver look up
#FACE_TURN_RIGHT	Driver turn head righ
#FACE_TURN_LEFT	Driver turn head left
#FACE_ID_ON	Measurements calculation begin
#FACE_NULL	No face identified
#REGIST_OK	Success driver registration
#REGIST_OTHER	Another driver name
#REGIST_OTHER_USE	Another driver name (name chosen already in use)

TABLE 4.5: TCP Command sended from local system

Commands sent from server to local system are very few. To activate the local system, the only necessity is to send a request to perform a recognition (first command), and then another command is sent with the associated name in order to save the acquired measurements on the database, and in case the local system does not recognize the driver, another command cancels any procedure, or just the request that was made to the local system. The commands can be seen in the table 4.6.

TCP Command	Fuction
#INIT_FACE_ID	Iniciate a facial recognition
#FACE_REGIST_name	Regist driver measurements with "name" chosen
#CANCEL_FACE_ID	Cancel any request

TABLE 4.6: TCP Command sended from server

### 4.2.3 GUI Application

Making a reference to the section 3.7.3, in order to remember the architecture defined for the GUI, it is possible to verify that the GUI was organized into 3 threads. The "Image Thread", which could also be called "Main Thread", would have the objective of updating the GUI image and labels, the "Button One Thread" that would initialize the driver's recognition process, and the "Button Two Thread" that would register the measurements made to the driver, in the database.

#### Image Thread

```

...
1  Bitmap colorBitmap;
   PXCImage.ImageData colorData;
   sample.color.AcquireAccess(PXCImage.Access.ACCESS_READ,

```

```

        PXCImage.PixelFormat.PIXEL_FORMAT_RGB24, out colorData);
2   colorBitmap = colorData.ToBitmap(0, sample.color.info.width, sample.color.info.height);
    ...

3   Graphics myGraphics = Graphics.FromImage(colorBitmap);
    using (Font arialFont = new Font("Arial", 3))

        for (int i = 0; i < faceLandmarkData.QueryNumPoints(); ++i) {
            int x = (int)points[i].image.x + LANDMARK_ALIGNMENT;
            int y = (int)points[i].image.y + LANDMARK_ALIGNMENT;

            string facepoint = "x"; //point without index

4       myGraphics.DrawString(facepoint, arialFont,
                               new SolidBrush(System.Drawing.Color.Pink), x, y);
    }
    ...

```

LISTING 4.18: Printing Landmark points

To make the image update, first it is necessary to acquire it from the camera, in a format in which it can be presented. The object used in C# to manipulate and represent images is the "Bitmap". For the image acquisition, an instance of this object, "colorBitmap", is created, and a request is made to the camera of an image format that can be saved in this instance, which is the format "PIXEL\_FORMAT\_RGB24", as can be seen in instruction 2, of the code previously presented.

After these instructions the image is ready to be presented, but for debug means and understanding of what is happening in the local system, one feature was added to the image: presenting the landmarks of the face that will be recognized.

In instruction 4, the image is converted into a format where it can be altered to print shapes or text, to better represent the landmarks.

The placement of the landmarks in the acquired image, of the camera, is performed in the instruction 4. By removing the traces in the image 4.5, it becomes possible to see the result of the printing of the landmarks.

```

private void UpdateUI(Bitmap bitmap)
{
    this.Dispatcher.Invoke(System.Windows.Threading.DispatcherPriority.Normal, new Action(delegate
    {
        // Display the color image
        if (bitmap != null) {
1       imgColorStream.Source = ConvertBitmap.BitmapToBitmapSource(bitmap);
        }

        // Update UI elements
2       lblNumFacesDetected.Content = String.Format("Faces Detected: {0}", numFacesDetected);

        if (UserName == null) lblUserName.Visibility = Visibility.Hidden;
        else {
            lblUserName.Visibility = Visibility.Visible;
3           lblUserName.Content = String.Format(UserName);
        }

4       if (UserName == "None User Identified") {
            button_Regist.IsEnabled = true;
            textBox.IsEnabled = true;
        }
        else {
5           button_Regist.IsEnabled = false;
            textBox.IsEnabled = false;
        }
6       if (lblWarnings_Visibility) {

```

```

        lblWarnings.Visibility = Visibility.Visible;
        lblWarnings.Content = lblWarnings_content;
        lblWarnings.Foreground = new SolidColorBrush((System.Windows.Media.Color)System.
            Windows.Media.ColorConverter.ConvertFromString(lblWarnings_color));
        if(lblWarnings_color == "#FF00FF00") button_Start.IsEnabled = true;
        else button_Start.IsEnabled = false;
    }
7   else {
        lblWarnings.Visibility = Visibility.Hidden;
        button_Start.IsEnabled = false;
    }

    // Change picture border color depending on if user is in camera view
8   if (numFacesDetected > 0) {
        bdrPictureBox.BorderBrush = System.Windows.Media.Brushes.LightGreen;
    }
9   else {
        bdrPictureBox.BorderBrush = System.Windows.Media.Brushes.Red;
    }
    });

    // Release resources
    bitmap.Dispose();
}

```

LISTING 4.19: Update Image in GUI

In the GUI image update, many actions occur and many features have been implemented.

In instruction 1 of the code previously presented, it is possible to see the placement of the above instance, "bitmap", in the location of the GUI intended for the presentation of a image. One feature added to the GUI was the fact that it made the presence of a face in the image, very evident. In instruction 8 and 9 it is implemented a rectangle printing around the image. If no face is identified in the image the rectangle is red, otherwise it is green.

In instruction 2, an update is performed on the label that indicates how many faces are on the image, and in instruction 6 the update of the label that indicates the calibration instructions of the face is made.

In instruction 4, the feature of only allowing the "Regist" button pressing after an attempt to recognize the friver has been implemented, the response from the local system is null. Not only will it be possible to press the button in this case, but also to write in the textbox destined for the driver's name by which they want to be recognized.

## Button One Thread

```

private void Star_button(object sender, RoutedEventArgs e)
{
    measure_process = true;
    aTimer.Enabled = true;
    UserName = null;

    lblUserNamePossible.Content = String.Format("-");

    for (int i = 0; i < size_measures; ++i) {
        currmeasure[i] = 0;
    }
}

```

LISTING 4.20: Star Identification Button routine

As previously mentioned, "Button One Thread" represents the function that is executed when the "Start Identification" button is pressed.

As in the case described in section 4.2.1, when the button is pressed the variable that allows the calculation of the measures on the driver face, "measure\_process", is set to "true", and once again the timer that defines how long the measurements are made, "aTimer", is started and the array that will contain the measurements averages made to the driver for 1 second, is initialized, that way preventing any vestige from previous recognitions from existing (extract code above).

### Button Two Thread

```
private void Regist_button(object sender, RoutedEventArgs e)
{
    String UserNameInsert = textBox.Text;
    Regist(UserNameInsert);
}
```

LISTING 4.21: Star Identification Button routine

As in the previous case, "Button Two Thread" represents a function executed when a button is pressed, but in this case the button "Regist". Since the function described in the section 4.2.1 performs already the analysis of the name chosen by the driver, in this function it is only collected the text/name written in the textbox and sent as a parameter, thus executing the instruction "Regist (UserNameInsert)", where "UserNameInsert" contains the text of the textbox.

```
private void Click_textBox(object sender, System.Windows.Input.MouseEventArgs e)
{
    textBox.Text = "";
}
```

LISTING 4.22: TextBox Update routine

A small feature has also been added to the textbox. If it is not clicked, its content will always be "(New User Name)", but if it is clicked, its content changes to null. This happens to prevent the driver from having to erase the contents of the textbox when writing the desired name.





# Chapter 5

## Results and Validation

In chapter five, the results of the design made in chapter three, and the implementation in four chapter, are described.

Initially in this chapter, a small performance test is presented between the libraries described in the section 3.6.4, followed by the graphical results obtained in the implementation of the GUI.

Also in this section it is demonstrated the result that can be seen from the implementation of the communication protocol described in the previous section and what can be found in the database used by the final system.

Here the performance tests to the final system developed are presented to facilitate the drawing of conclusions for the next chapter.

### 5.1 Initial Tests

To justify the choice of the library function described in section 3.6.4, a small test was performed. In this test the challenge was to acquire an image, identify a face and present the FPS rate.

In order to maintain the conditions equal in each case, the identification was limited for only one face, and the image presented would have to contain a rectangle around the identified face.

The resulting applications, in each implementation required to respond to the challenge, were executed on the same PC, that way, the same processing power could be provided in each case.

The result can be seen in the following sections.

### 5.1.1 OpenCV



(A) Face identified

```

Frames per second: 1
Frames per second: 7
Frames per second: 16
Frames per second: 17
Frames per second: 16
Frames per second: 16
Frames per second: 17
Frames per second: 16
Frames per second: 17
Frames per second: 17
Frames per second: 14
Frames per second: 16
Frames per second: 16
Frames per second: 16
Frames per second: 18
Frames per second: 15
Frames per second: 17
Frames per second: 17
Frames per second: 18
Frames per second: 16
Frames per second: 16
Frames per second: 6
Frames per second: 17
Frames per second: 19
Frames per second: 16
Frames per second: 16
Frames per second: 13

```

(B) FPS Rate

---

FIGURE 5.1: OpenCV results to the challenge

For the development of the algorithm needed to obtain what is described above, Visual Studio was used.

Since the camera selected in the section 3.5.2 does not support the use of the functions of OpenCV library, another camera was used. The chosen camera to run the test applications was the one that was already integrated in the PC. The characteristics of this camera are identical in resolution level and maximum FPS at each resolution, to the chosen camera.

The acquisition of a 720p resolution image was defined, and both cameras support a 60 FPS image acquisition rate in this resolution.

In the figure 5.1a, it is possible to see that the algorithm developed, using the OpenCV library, is able to identify a face, and based on the output of its functions, it was possible to place a rectangle on the identified face. One of the problems that can be identified is that the rectangle does not cover the whole face, but only part of it.

In the second figure, 5.1b, it is presented the image acquisition rate per second that can be obtained using this library. The maximum that could be obtained in this case was 19 FPS.

## 5.1.2 Intellisense

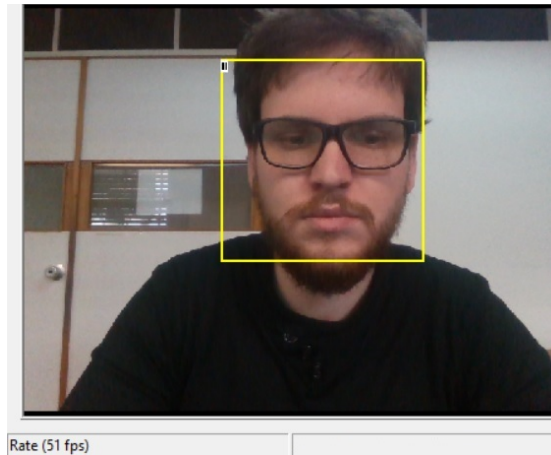


FIGURE 5.2: Intellisense results to the challenge

In the above figure it is possible to see the result to the test, when the library Intellisense is used.

Compared to the previous case, the square representing the identified face cover the entire face, thus demonstrating much more accurate and reliable results. It is possible to see in the lower left corner of the image, the image acquisition rate in this case. Compared with the previous case, it is possible to see an increase of almost 270% of more frames. This increase of frames allows the presentation of a more fluid stream and an acquisition of a larger amount of data "of the images" in less time, since in the previous case it could only be acquired the data of 19 images in a second, instead in this case the data from 51 images can be acquired in one second.

The choice for the library of functions to use was obvious, making it possible the use of the camera chosen in the section 3.5.2.

## 5.2 GUI Application

In order to perform the tests on the GUI features, the GUI was first analyzed to see if it contains all the components described in the previous chapters, and then performed the tests to analyze if the features, and their main fuction, work. Since the local system does not contain a visible output, it will be through the GUI output that the verification will be made.

### 5.2.1 GUI Components

The first test was the analysis of the components that compose the GUI. In the figure 3.22, it is possible to see the design expected for the GUI, as well as the expected components that it has. In the figure 5.3a, it is possible to see the graphic view of the GUI and in figure 5.3b the same figure but with the components numbered.

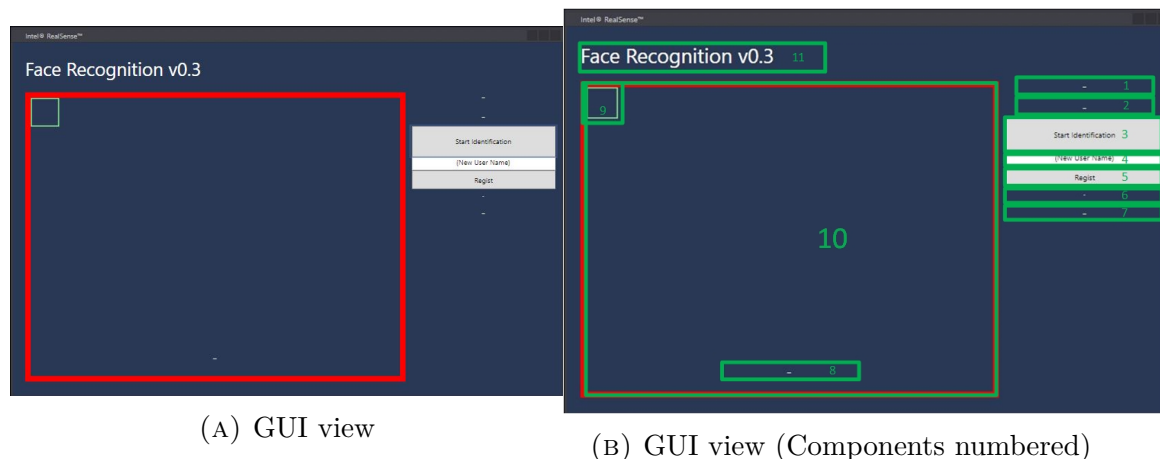


FIGURE 5.3: GUI components analyze

**Results:** An analysis of the GUI indicates the presence of all the components described in figure 3.22, since it contains everything that was described in the design phase, and more. A numbering of the components was made in the figure 5.3b, making it possible to transcribe to a table each GUI component, as well as their function (table 5.1).

Component	Fuction
1(label)	Number of faces identified
2(label)	Additional information
3(button)	Start recognition trigger
4(textbox)	New driver name
5(button)	Make a driver regist trigger
6(label)	Regist process status
7(label)	Driver name
8(label)	Face position indications
9(Square form)	Face presence indicator
10(image space)	Camera streaming + landmark positions
11(title)	GUI name

TABLE 5.1: GUI components

### 5.2.2 GUI Connections Warnings

One of the needs for the correct functioning of the local system is for the camera to be connected and for its remote activation, the TCP server to be active. To inform the user, who is handling the local system by the GUI, that some of the previous situations do not occur, a warning has been implemented in the GUI. The first test consisted of activating the local system without the camera being connected, and in the second test, the TCP server was not active.

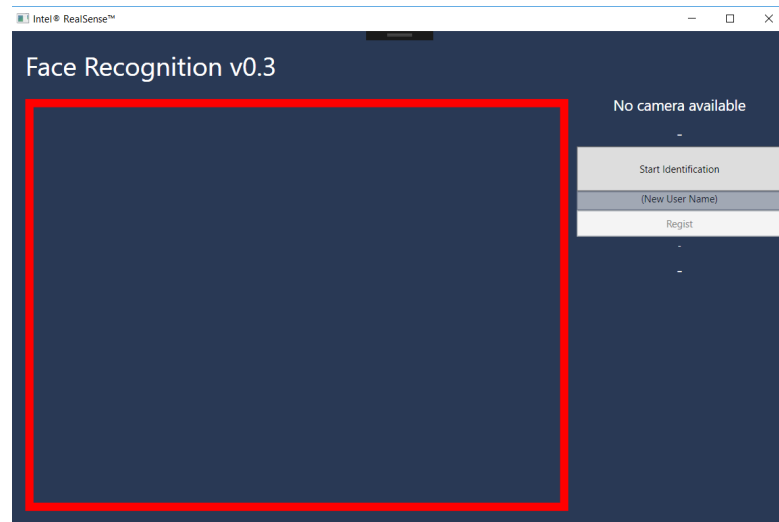


FIGURE 5.4: Camera not connected (Warning)

**First test results:** As it is possible to verify in the figure 5.4, a warning was presented due to the fact that the camera is not connected to the local system. The only way to solve this problem is to "kill" the local system process, that is, shut down the local system process and restart it, with the camera connected to it.

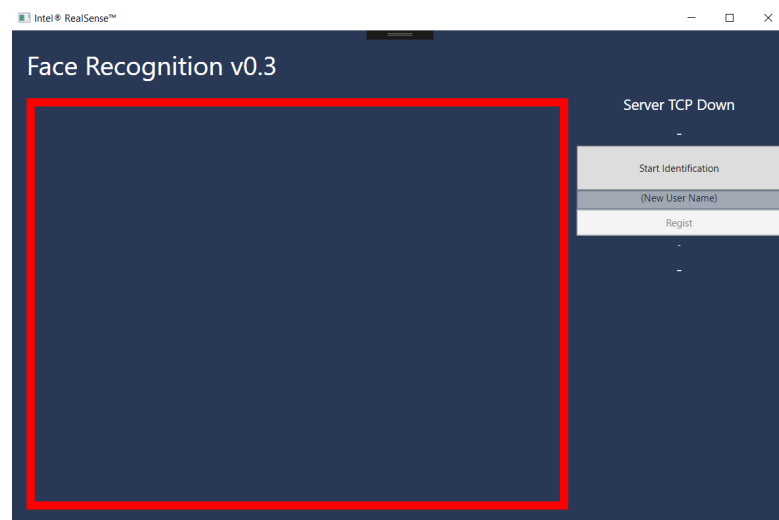


FIGURE 5.5: TCP connection not made (Warning)

**Second test results:** The fact that the TCP server is not active when the local system starts is not problematic, because the local system can achieve its main objective without its need. The fact that it is not connected will only interfere with the activation of the local system's main objective remotely. This warning was implemented to make it easier to integrate with DSM and respond to issues in the TCP connection.

### 5.2.3 GUI/Local System Normal Cycle

For the verification that the local system meets the purpose for which it was created, a series of tests have been performed. The tests were aimed at testing all phases of the local system algorithm and verifying that the GUI presented all the information that occurred. The tests performed are:

- Attempt to recognize a new user;
- Activation of the possibility of a registration;
- Making a registration;
- Attempt to recognize a user already in the database.

To perform these tests, it was firstly requested to the local system to try recognizing a user that was not in the database. As it is expected for the system to not get a match, it is expected to unlock the possibility of registration in the GUI. After writting a name, the registration for the new user is made. Once registered, local system will try to recognize the same user, returning the previously chosen name, as the user ID.

#### New user recognition

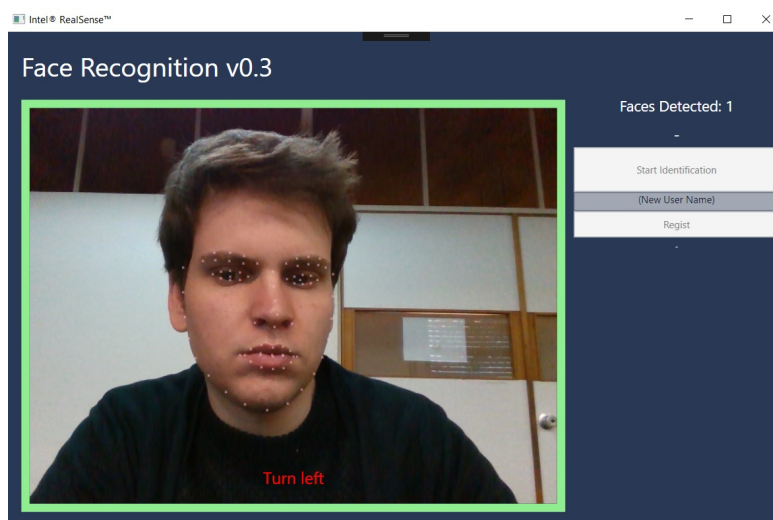


FIGURE 5.6: Face position indications

**Results:** As expected, for the recognition of any user, it is needed that the face is in the required position to make the measurements calculation. As can be seen in figure 5.6, the indications necessary for the user to correct their position are being displayed, and the button for triggering the recognition is not active, and registration button is not active as well.

If all conditions are followed, it is expected that the button "Star Identification" becomes active, and as can be seen in the figure 5.7, the expected happens. After all the indications followed, the green message "Scan available" appears in the place where

the instructions are indicated, and the "Star Identification" button becomes active, in order to make a recognition possible, if desired.

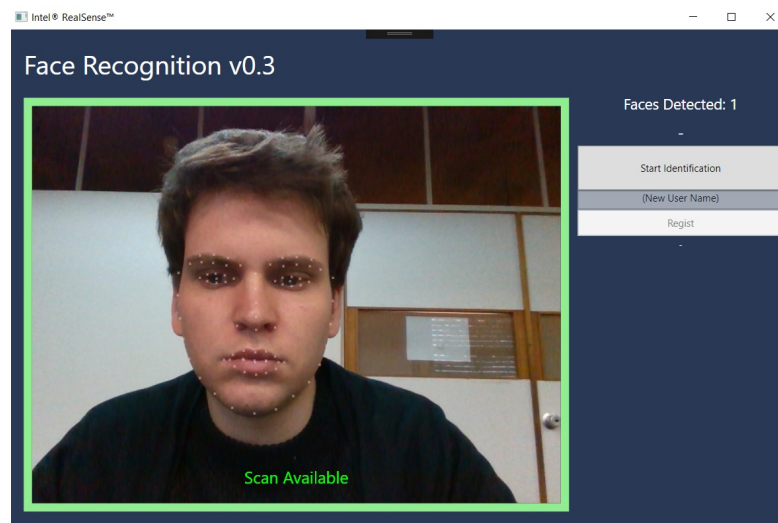


FIGURE 5.7: Face recognition available

**Results(cont.):** With all conditions verified, the recognition is realized, and as it can be seen in figure 5.8, the system does not recognize the user. As was indicated in the test conditions already described, this recognition attempt was performed on an unidentified user, ie, the result expected was that no identification of the same is returned, as it happened.

In the figure 5.8, it is possible to verify also that in the stream of the image captured by the camera, the landmarks of the user face are represented. This representation will only be made in the user that the system will recognize, that is, if more than one face is captured, only one of them will see their landmarks represented.

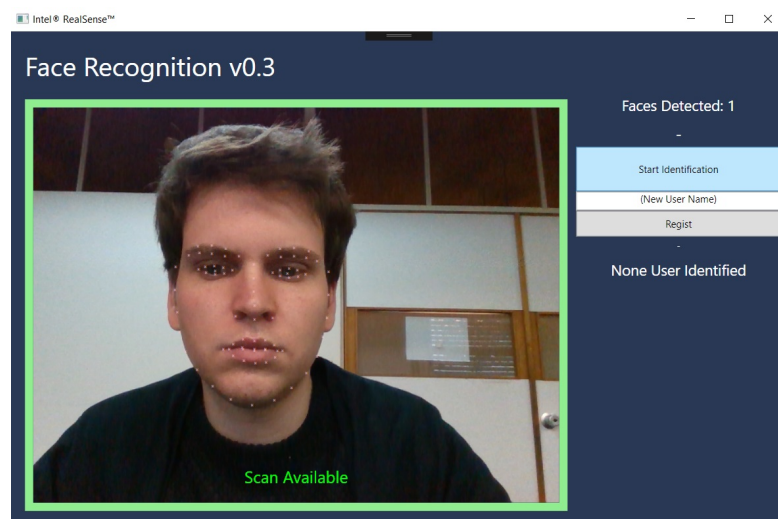


FIGURE 5.8: Face recognition of a new user



## Registration activation

**Results:** The figure 5.8 corresponds to one of the tests performed on the implemented system. As expected from a recognition attempt that did not return the user ID, in the figure presented, it is possible to see that it is now possible to write a name for the association with the measurements of the new user, and the trigger to perform the registration, is also active ("Regist" button). For the realization of the registry it is not necessary for the user to be in a specific position, because his measurements already have been calculated in his attempt of recognition.

## Registration attempt



FIGURE 5.9: Writing the user's name

**Results:** In this test the possibility of writing a name/ID to the new user was expected. As it can be seen in the figure 5.9, it is possible to write something in the textbox destined to assign a ID to the new user and to make the following tests, the name "Barbosa" was chosen.

After the name is chosen,, the "Regist" button is pressed to perform the registration. If a user already has the same ID, a warning will appear, but as can be seen from the figure 5.10 the ID user was successfully registered and with this, everything returns to the initial state. The button "Start Identification" is not active because the user in the image is not in the correct position, and the textbox and "Regist" button are disabled to prevent a second registration associated to the same user.





FIGURE 5.10: User registration

### Known user recognition



FIGURE 5.11: Face recognition available again

**Results:** After the user is registered, a new attempt to recognize the user is performed. As can be seen from the figure 5.11, the user again followed the instructions to position his face and thus make a new recognition.

Once the recognition is made, the expected is for the local system to return the ID name "Barbosa". As can be seen in the figure 5.12, the local system, after the recognition attempt, the ID name "Barbosa" is returned, thus identifying the user that was in the image.

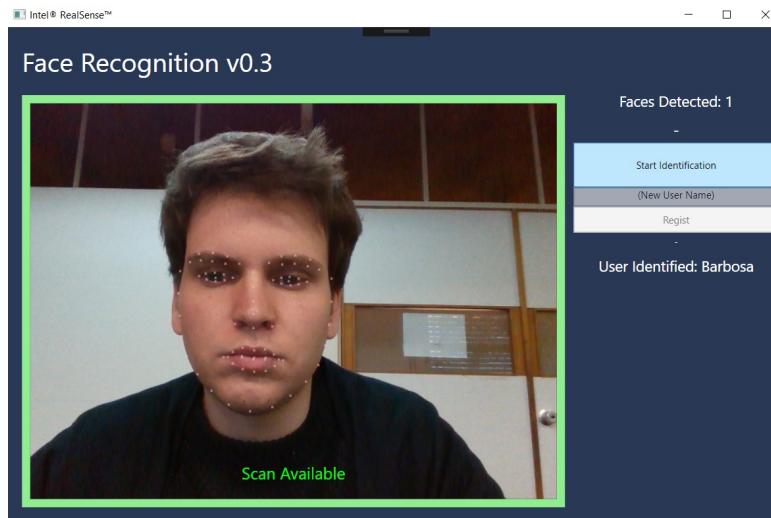


FIGURE 5.12: Face recognition of known user

### 5.2.4 GUI/local System Multi Face

Another test performed was verifying if the system was able to identify a second face in the image acquired by the camera, and if it really would only represent the landmarks of the face that was meant to be recognized.

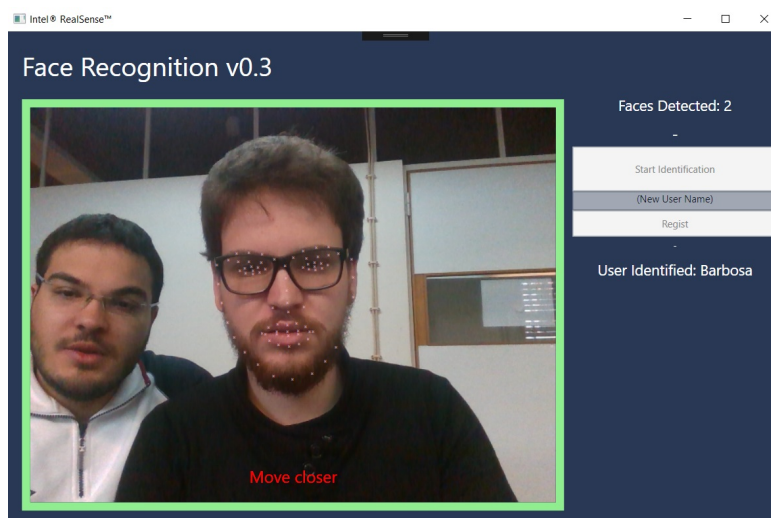


FIGURE 5.13: Multi face detection

**Results:** The result of this test demonstrated that the system is able to identify more than one face, as can be seen in the figure 5.13. The system identified more than one face, as described in the first information represented on the right side of the GUI, and just represented the landmarks of the recognized face, if an request was received. The face to be recognized is chosen based on the distance from the camera, and as can be seen, the face chosen was the one closest to the camera.

## 5.3 Server System

Since it is a key to the correct functioning of the local system, tests were also made on the server to confirm its ability to respond to multiple clients, as well as to communicate with the local system using the communication protocol described in section 4.2.2.

### 5.3.1 Multiclient Handling

As described in the section 3.7.2, the server system was defined and implemented to be capable of handling several TCP clients at the same time, and for this, in its implementation the multi-thread approach was used, where to each client connected a process, independent of the rest the others, is created in order to execute all in parallel to avoid the loss of information by a TCP client.

For this test the "Telnet" command was used on the Linux OS command line. This command creates a remote virtual connection with a server, only being necessary to provide the IP of the server and the port. The "Telnet" command has a format close to the following: "telnet IP\_server Port\_server".

Two OS Linux command lines were used, and the command "Telnet" was used to create a connection to the server system. To test that the server was able to receive messages from both clients, some test unique messages were sent from each clients.

```

qtcreator_process_stub
Listening to port: 5000
Waiting for connection...
Connecting
***
Client is connected
Data received from client: "Test message from client 1"
Connecting
***
Client is connected
Data received from client: "test message from client 2"
Data received from client: "Second message from client 1"
Data received from client: "Second message from client 2"
[]

pedro@pedro-virtual-machine: ~
pedro@pedro-virtual-machine:~$ telnet 127.0.0.1 5000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
Test message from client 1
Second message from client 1
[]

pedro@pedro-virtual-machine: ~
pedro@pedro-virtual-machine:~$ telnet 127.0.0.1 5000
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
test message from client 2
Second message from client 2
[]

```

FIGURE 5.14: Server multiclient handling

**Results:** The test performed on server system was a success, because as it is possible to analyze in the figure above, both TCP clients were able to connect to the server system without problems, where each one sent a message when they made their connection and the server received both, sending a message to each client at the same time with a printing of those messages appearing in the server output, proving the multiclient handling.

### 5.3.2 Communication Protocol

For the test of the communication protocol, two situations were created: a recognition request to the local system of a user that was already in the database, and another

recognition request of a user that was not in the database. The results can be seen in the following sections.

### Known user recognition

Taking advantage of the test already presented in the section 5.2.3, a request was made to the local system to recognize the same user, expecting the result to be the message "ID\_FACE\_Barbosa", which means, "The identified user has the ID Barbosa".

```
Listening to port 5000
Waiting for connection...
Connecting
***
Client is connected
Data received from client: "#FACE_RECOG_ON"
#INIT_FACE_Id
#INIT_FACE_Id
Data sented from server: "#INIT_FACE_ID"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_TURN_UP"
Data received from client: "#FACE_ID_ON"
Data received from client: "#ID_FACE_Barbosa"
```

FIGURE 5.15: Protocol to known user

**Results:** By the figure presented, it is possible to verify that the test was a success, not only were the messages sent and received correctly in the local system and the server system, but also that the user was successfully recognized and sent its ID with equal success.

When the local system initiates, a message informing the server is sent, and that message was sent successfully. As expected, when the server sent a recognition request to the local system, this started sending calibration messages to the server. Then the server was informed that the "Recognition Process" was in progress, followed by the message that informs the user's identity, "ID\_FACE\_Barbosa", and that means the user has been identified with the "Barbosa" ID.

### Unknown User Recognition

With the test of handling messages from a request to a user known to the system performed, a second test was performed in order to test the messages coming from an unknown user. The expected in this situation is the following local system response message: "ID\_FACE\_NONE", which means that the user has not been recognized, and the server system must send the ID of this new user. At the end a confirmation message from the user's registration should be received.

```

#INIT_FACE_ID
Data sended from server: "#INIT_FACE_ID"
Data received from client: "#FACE_MOV_CLOSER"
Data received from client: "#FACE_TURN_UP"
Data received from client: "#FACE_ID_ON"
Data received from client: "#ID_FACE_NONE"
#FACE_REGIST_BarBoSa
Data sended from server: "#FACE_REGIST_BarBoSa"
Data received from client: "#REGIST_OK"
#INIT_FACE_ID
Data sended from server: "#INIT_FACE_ID"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_MOV_CLOSER"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_MOV_CLOSER"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_MOV_CLOSER"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_MOV_CLOSER"
Data received from client: "#FACE_TURN_LEFT"
Data received from client: "#FACE_ID_ON"
Data received from client: "#ID_FACE_BarBoSa"

```

---

FIGURE 5.16: Protocol to unknown user

**Results:** The figure 5.16 represents the test result described, and can be concluded that this is another successful test. Unlike the previous case, an initial local system message was not sent because this test was performed immediately after the previous one. Once again the local system sent the calibration messages, but this time the answer to the recognition attempt was "ID\_FACE\_NONE", as expected, activating the process of assigning an ID to the user.

The server sent the message "FACE\_REGIST\_BarBoSa", which means: "register the user with the ID name BarBoSa" and the local system's response was "REG-IST\_OK", registration complete with success. In order to verify that the registration was successful, a new user recognition request was made to the local system, and the response this time was "ID\_FACE\_BarBoSa", as expected the user was identified.

## 5.4 Local System

With all systems adjacent to the main system tested, it is time to perform tests to the local system. These tests aim to capture information about the probabilistic characteristics of the system, that is, some variables were varied, such as the number of users in the database, the number of measures calculated, among others, in order to see if there are changes in the accuracy percentage of the local system to identity the user.

The results of these tests, can be see in table form, where it is presented the characteristics changes, as well as the probability of the system recognizing the face on the image captured by the camera. For each situation 10 recognition attempts were made and the results were analyzed.

Remembering, the standard conditions that the system has until the follow tests were:

- Users database number: Undefined
- Measurements number: 20
- Measurements interval (mm): -2.5 to 2.5
- Equal measurements percentage: 75
- Update measurements: On

To the test, only the characteristics in study were changed, keeping the others unaltered.

### 5.4.1 Users Database

Number of Users in Database	System Accuracy (%)
1	98
2	96
3	94
4	92
5	92

TABLE 5.2: System accuracy based on users database

**Results:** In this test the system was able to always recognize the user, but to calculate the accuracy of the system it has to take into account variables that do not have control, such as light reflection in the glasses, intense illumination or unexpected movement during the recognition process.

### 5.4.2 Measurements

Measurements Number	System Accuracy (%)
3	0
5	20
10	50
15	85
20	92

TABLE 5.3: System accuracy based on measurements calculated

**Results:** As expected, as indicated by the table 5.3, with the increase in the number of measurements calculated from the user image face, the system becomes more precise.

### 5.4.3 Search Interval

Measurements Interval (mm)	System Accuracy (%)
0	0
-0.5 to 0.5	25
-2.5 to 2.5	92
-4.5 to 4.5	78
-6.5 to 6.5	33

TABLE 5.4: System accuracy based on measurements interval

**Results:** From the results of these tests two conclusions was taken: the interval can not be too small, since the calculated measurements will never be exactly the same, and the interval can not be too large or else many users will be compatible with the measurements taken.

### 5.4.4 Compatible Percentage

Equal measurements (%)	System accuracy (%)
50	55
65	70
75	92
80	80
90	10

TABLE 5.5: System accuracy based on identical measurements

**Results:** This is one of the most important tests, because it is possible to conclude that when there is a very low number of compatibilities, the result can be many users, but if it is also very high, there can be no correspondences, because if the user change their expression, the measurements taken will change and the compatibility with the user's original measurements will be very low.

### 5.4.5 Update Measurements

Update measurements	System accuracy (%)
On	92
Of	65

TABLE 5.6: Update measurements after recognition

**Results:** the implementation of this function was to perform updates as a means of correcting small changes of the driver's face measurements, with the passing of time, thus being a means of verification of how the luminousness affects the landmarks. During the tests, it was noted that extreme lighting slightly modifies the coordinates of the landmarks, leading to recognition failures. This implementation corrected this situation.

## 5.5 Database Content

Not with the purpose of testing something but simply verify, the contents of the database were registered at the end of the previous tests, in order to verify that all the users were correctly saved and the information correctly grouped.

UserName	Measure1	Measure2	Measure3	Measure4	Measure5	Measure6	Measure7	Measure8	Measure9	Measure10	Measure11	Measure12	Measure13	Measure14
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1 Barbosa	14.62591133...	92.67168130...	68.04668065...	58.30869084...	44.78087624...	41.06217272...	102.4514613...	113.5061148...	36.11810257...	39.91598922...	34.49149630...	49.31642590...	34.35107809...	61.35534175
2 Pedro	17.47696358...	90.95309532...	71.77337432...	57.76079297...	48.23431104...	41.79160286...	104.3754579...	112.8473839...	37.46074392...	38.98341083...	30.11337672...	49.13127257...	35.45000284...	59.27233309
3 Artur	37.06975816...	99.62722739...	86.66340132...	65.14808640...	86.49987631...	47.11899167...	126.6110680...	129.1034066...	44.73199509...	45.65501635...	39.33957925...	56.34265676...	36.69891716...	62.91664812
4 Antonio	22.34872207...	87.30728271...	70.85849110...	53.54736574...	50.81570093...	39.12534196...	107.1581387...	109.6458928...	35.12782651...	37.99978656...	32.51771214...	50.66650672...	35.23510134...	57.39611802
5 ArturSemOcul...	32.80289188...	96.09001312...	80.38755789...	61.68108233...	62.04633346...	43.89530713...	126.0458610...	127.2739958...	38.45976157...	40.46096909...	36.88402118...	62.22407960...	37.84433952...	65.32692161

FIGURE 5.17: Database contest after the tests

**Results:** All the information is correctly found in the database, that is, another test with positive results.



# Chapter 6

## Conclusions and Future Work

In this chapter, a reflection is made of all the work made to complete this project. In this reflection the project conclusions are discussed, as well, proposals for future work, which are equally important. In this proposal ideas, or implementations are presented, that can increase the efficiency of the project developed in this dissertation.

### 6.1 Conclusions

The system developed in this dissertation consists on a algorithm capable of recognizing drivers inside their car, as well as loading the costumizations the drivers made to their vehicle each time they use it.

With the implemented system, it is possible to solve the problem of loading profiles saved in the car memory, where each profile stored characteristics such as seat and steering wheel height, central stack colors, among others. This way the driver avoids the creation or choice of a profile, just needing to look at a camera and expect to be recognized. The solution presented here only requires a specific camera, a Mini PC and a connection with the central system of the vehicle.

The system requires that the user has to remain immobile for one second, in a specific position, to the conditions maintain equals all the time. A situation that could not be solved in the development of this dissertation, was the fact that it is not possible to attribute the same identity to the same person with, and without, glasses, that is, the system will create two identities of the same user, one when using glasses and another one without glasses.

The developed system contains an offline database, ie, in case of internet failure connection, the system is still able to recognize the driver, due to its independence with the exterior. It is also possible to use the system without it being connected to a central system of a vehicle, due to a graphic solution also implemented.

To communicate with the outside world, the system was incorporated with a GUI, to present the acquired image, as well as to realize recognitions and registries in the database. The system is also built with a TCP connection, so that it can connect to a TCP server, and allow the server to send requests for recognitions or user registrations.

Finally, in conclusion, the requirements that are presented in chapter 1.5 were almost filled entirely, just one of them does not allow to say that all were achieved. The "Image Acquisition" requirements were achieved when a new 3D image was used for the driver's recognition having no physical characteristics and it was implemented an algorithm to minimize a specific driver position to reduce the error occurrence. In "Data processing" part, the requirements were achieved since the final system is compact, if installed on Intel NUC PC, and allow the driver with a bigger accuracy of 75%, in only 1 second. The new system is able to compare all the new users with drivers that are already part

of the data base, This way verifying the requirements "Data Comparasion". Since it allows its activation with TCP/IP commands, and the sending of the driver's identity, the system also verifies the "Server Communication" requirements. In the "System" requirements, only can be stated that only one was not successfully concluded, because of the system installation was made in a normal sized PC, this was not made compact and of easy integration on DSM. So, only when Intel NUC PC component is acquired and the final system is implemented on it, the system will be answer completely to the "System" requirements.

## 6.2 Future Work

The system implemented in this dissertation still lacks some improvements, from new functionalities, as well as making its integration in a small processing hardware easier.

As a first suggestion, it is advisable to purchase the Intel NUC Mini PC equipment to reduce the size of the local system has at this moment.

The second suggestion would be to remove all restrictions on face position and recognition. One solution would be the use of the HMM's or the use of one of the IntelSense camera SR300 features, the 3D scan.

As a third suggestion, it would be creating a shared database and sending the image acquired by the local system, to the TCP server. This would allow the car's system to record photos or driver stream while driving for other features, such as sending the stream to assistance units in the event of an accident. Sharing the database would allow both the local system and the server to register new drivers without requiring requests between them.

For fourth suggestion, making IR light interference tests on 3D image acquisition and during the 3D scan, of the camera. This would allow a better understanding of the limitations of the camera chosen.

And as a last suggestion, exploration of the other functionalities of the chosen camera, for possible integration with the current local system, since the selected camera was not limited in its creation, only to the objective for which it was used in this dissertation, but many more like, Hand Tracking, Eye Tracking, Speech Recognition, among others can be a valid option.

# Bibliography

- [1] “4-megatrends-shaping-the-automotive-and-car-market @ www.consultancy.uk.” [Online]. Available: <http://www.consultancy.uk/news/2609/4-megatrends-shaping-the-automotive-and-car-market>
- [2] “driver\_profile @ www.bmw.com.” [Online]. Available: <http://www.bmw.com/com/en/insights/technology/technology{ }guide/articles/driver{ }profile.html?source=categories{&}article=driver{ }profilehttp://www.bmw.com/com/en/insights/technology/technology{ }guide/articles/driver{ }profile.html?source=categories{&}article=driv>
- [3] “lebih-dari-2000-rfid-tags-kini @ insaviewpoint.blogspot.pt.” [Online]. Available: <http://insaviewpoint.blogspot.pt/2006/03/lebih-dari-2000-rfid-tags-kini.html>
- [4] C. Center, “Mathematical Methods of Pattern Recognition,” RUSSIAN ACADEMY OF SCIENCES COMPUTER CENTER, Tech. Rep., 2005. [Online]. Available: <http://www.mmro.ru/files/2005-mmro-12.pdf>
- [5] “biometric @ www.physionet.org.” [Online]. Available: <http://www.physionet.org/pn3/ecgiddb/biometric.shtml>
- [6] “fingerprint-matching-in-mobile-devices-android-platform @ answers.opencv.org.” [Online]. Available: <http://answers.opencv.org/question/6364/fingerprint-matching-in-mobile-devices-android-platform/>
- [7] “US7602947 @ www.google.com.” [Online]. Available: <https://www.google.com/patents/US7602947>
- [8] “infrared @ itleivrl.epfl.ch.” [Online]. Available: <http://ivrl.epfl.ch/research/infrared>
- [9] “infrared-light @ learn.sparkfun.com.” [Online]. Available: <https://learn.sparkfun.com/tutorials/light/infrared-light>
- [10] “3-d-imaging-advances-capabilities-of-machine-vision-part-i @ www.vision-systems.com.” [Online]. Available: <http://www.vision-systems.com/articles/print/volume-17/issue-4/departments/leading-edge-views/3-d-imaging-advances-capabilities-of-machine-vision-part-i.htmlhttp://www.vision-systems.com/articles/print/volume-17/issue-4/departments/leading-edge-views/3-d>
- [11] “understanding-laser-based-3d-triangulation-methods @ www.vision-systems.com.” [Online]. Available: <http://www.vision-systems.com/articles/print/volume-20/issue-6/features/understanding-laser-based-3d-triangulation-methods.html>
- [12] “index @ www.depthbiomechanics.co.uk.” [Online]. Available: <http://www.depthbiomechanics.co.uk/?p=102>

- [13] “3d-a-closer-look-at-time-of-flight-technology @ icdsecurity.com.” [Online]. Available: <http://www.icdsecurity.com/2013/08/30/security-in-3d-a-closer-look-at-time-of-flight-technology/>
- [14] “4347-1065nm-and-1310nm-mems-tunable-vcsel-light-source-technology-for-oct-imaging @ spie.org.” [Online]. Available: <http://spie.org/newsroom/4347-1065nm-and-1310nm-mems-tunable-vcsel-light-source-technology-for-oct-imaging?highlight=x2404{&}articleid=x88950http://spie.org/newsroom/4347-1065nm-and-1310nm-mems-tunable-vcsel-light-source-technology-for-oct-imaging?highlight>
- [15] R. Jafri and H. R. Arabnia, “A Survey of Face Recognition Techniques,” *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.
- [16] Z. Biuk and S. Loncaric, “Face recognition from multi-pose image sequence,” *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat. No.01EX480)*, pp. 319–324, 2001. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=938649>
- [17] “basic @ vismod.media.mit.edu.” [Online]. Available: <http://vismod.media.mit.edu/vismod/demos/facerec/basic.html>
- [18] R. Brunelli and T. Poggio, “Face recognition: features versus templates,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993, pp. 1042–1052.
- [19] G. G. Gordon and P. H. Gb, “Face recognition based on depth maps and surface curvature,” *SPIE Geometric methods in Computer Vision*, pp. 234–247, 1991. [Online]. Available: <http://octopus.cs.drexel.edu/{~}gao25/References/Gordon{ }SPIE91{ }faceRecognition.pdf>
- [20] “volvo-automated-driver-monitoring @ newatlas.com.” [Online]. Available: <http://newatlas.com/volvo-automated-driver-monitoring/31257/>
- [21] “1024835\_Bmw-Using-Face-Recognition-To-Personalize-Cars @ Www.Motorauthority.Com.” [Online]. Available: <http://www.motorauthority.com/news/1024835{ }bmw-using-face-recognition-to-personalize-cars>
- [22] “1092960\_Ford-and-Intel-Use-Facial-Recognition-To-Improve-in-Car-Tech-Safety @ Www.Motorauthority.Com.” [Online]. Available: <http://www.motorauthority.com/news/1092960{ }ford-and-intel-use-facial-recognition-to-improve-in-car-tech-safety>
- [23] “chinarealtime @ blogs.wsj.com.” [Online]. Available: <http://blogs.wsj.com/chinarealtime/2016/03/26/chinese-researchers-invent-new-police-car-that-can-scan-criminals-faces/>
- [24] “f65aaf83395ccb9c8c2d969d7835f0a89389699ac @ www.slashgear.com.” [Online]. Available: <http://www.slashgear.com/roadrage-car-tech-tracks-if-youre-dangerous-14320681/>
- [25] “intelrealsensetm-developer-kit-featuring-sr300 @ click.intel.com.” [Online]. Available: <http://click.intel.com/intelrealsensetm-developer-kit-featuring-sr300.html>

- [26] “kinect-on-ps3 @ www.techpedition.com.” [Online]. Available: <http://www.techpedition.com/2011/03/20/kinect-hacked-to-work-on-ps3/>
- [27] “perception-cam-industrial-multi-dimension-camera @ www.matrix-vision.com.” [Online]. Available: <https://www.matrix-vision.com/perception-cam-industrial-multi-dimension-camera.html>
- [28] “nuc-kit @ www.intel.com.” [Online]. Available: <http://www.intel.com/content/www/us/en/nuc/nuc-kit-nuc6i5syh.html>
- [29] “poseeulerangles\_pxcfacedata @ software.intel.com.” [Online]. Available: [https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/poseeulerangles\\_{\\_}pxcfacedata.html](https://software.intel.com/sites/landingpage/realsense/camera-sdk/v1.1/documentation/html/poseeulerangles_{_}pxcfacedata.html)
- [30] S. Tikoo and N. Malik, “Detection , Segmentation and Recognition of Face and its Features using Neural Network,” vol. 5, no. 6, pp. 130–136, 2016.
- [31] “blog-global-automotive-industry-trends-forecasts @ www.ispiritsolutions.com.” [Online]. Available: <http://www.ispiritsolutions.com/blog-global-automotive-industry-trends-forecasts/>
- [32] “automotive-display-market-to-benefit-from-the-proliferation-of-internet-of-things @ technology.ihs.com.” [Online]. Available: <https://technology.ihs.com/503499/automotive-display-market-to-benefit-from-the-proliferation-of-internet-of-things>
- [33] “Power\_seat @ en.wikipedia.org.” [Online]. Available: [https://en.wikipedia.org/wiki/Power\\_{\\_}seat](https://en.wikipedia.org/wiki/Power_{_}seat)
- [34] “Seat-Memory @ cars.axlegeeks.com.” [Online]. Available: [http://cars.axlegeeks.com/d/f/Seat-Memory-\\_{%}28Driver\\_{%}29](http://cars.axlegeeks.com/d/f/Seat-Memory-_{%}28Driver_{%}29)
- [35] “index @ web.frotcom.com.” [Online]. Available: <https://web.frotcom.com/{#}/?ci=pt-pt>
- [36] A. Aparecida, “CONTRIBUTIONS TO BIOMETRIC RECOGNITION : MATCHING IDENTICAL TWINS AND LATENT FINGERPRINTS,” pp. 1–198, 2013. [Online]. Available: [http://biometrics.cse.msu.edu/Publications/Thesis/AlessandraPaulino\\_{\\_}ContributionsBiometricRecognitionIdenticalTwinsLatentFp\\_{\\_}PhD13.pdf](http://biometrics.cse.msu.edu/Publications/Thesis/AlessandraPaulino_{_}ContributionsBiometricRecognitionIdenticalTwinsLatentFp_{_}PhD13.pdf)
- [37] I. F. Orsag, “Biometric Security Systems,” *WO Patent WO2007128975*, p. 91, 2004. [Online]. Available: <http://www.fit.vutbr.cz/research/pubs/theses/orsag2004.pdf>
- [38] U. G. J. Balis and A. Arbor, “Robust Identification of the Patient : Biometrics Introduction : A General Discussion of Biometrics Biometric Measurement Modes ;,” 2013. [Online]. Available: <http://uscapknowledgehub.org/site{~}/102nd/pdf/companion04h01.pdf>
- [39] “PQRST @ www.cyberphysics.co.uk.” [Online]. Available: <http://www.cyberphysics.co.uk/topics/medical/heart/PQRST.html>
- [40] E. R. Sisco, “METHODS AND STANDARDS FOR THE ANALYSIS AND IMAGING OF LATENT FINGERPRINTS AND TRACE CONTRABAND USING AMBIENT IONIZATION MASS SPECTROMETRY AND SECONDARY

- ION MASS SPECTROMETRY,” 2014. [Online]. Available: [http://drum.lib.umd.edu/bitstream/handle/1903/15270/Sisco{}\\_umd{}\\_0117E{}\\_15070.pdf;jsessionid=562119E7822D7350999BAC158794CFD6?sequence=1](http://drum.lib.umd.edu/bitstream/handle/1903/15270/Sisco{}_umd{}_0117E{}_15070.pdf;jsessionid=562119E7822D7350999BAC158794CFD6?sequence=1)
- [41] “B002AUTV8K @ www.amazon.com.” [Online]. Available: <https://www.amazon.com/Biometric-Security-Fingerprint-Car-Starter/dp/B002AUTV8K>
- [42] B. H. Juang and L. R. Rabiner, “Automatic Speech Recognition ,À A Brief History of the Technology Development,” *Elsevier Encyclopedia of Language and Linguistics*, pp. 1–24, 2005. [Online]. Available: [http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354{\protect\T1\textbraceleft}{\\_}{\protect\T1\textbraceright}LALI-ASRHistory-final-10-8.pdf](http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354{\protect\T1\textbraceleft}{_}{\protect\T1\textbraceright}LALI-ASRHistory-final-10-8.pdf)
- [43] D. a. Reynolds and R. C. Rose, “Robust text-independent speaker identification using Gaussian mixture speaker models,” pp. 72–83, 1995. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=365379>
- [44] “irisrecognitiontechnology @ www.irisid.com.” [Online]. Available: <http://www.irisid.com/productssolutions/technology-2/irisrecognitiontechnology/>
- [45] “Image\_sensor @ en.wikipedia.org.” [Online]. Available: [https://en.wikipedia.org/wiki/Image{}\\_sensor](https://en.wikipedia.org/wiki/Image{}_sensor)
- [46] “Fotodiodo @ pt.wikipedia.org.” [Online]. Available: <https://pt.wikipedia.org/wiki/Fotodiodo>
- [47] “Pontos\_por\_polegada @ pt.wikipedia.org.” [Online]. Available: [https://pt.wikipedia.org/wiki/Pontos{}\\_por{}\\_polegada](https://pt.wikipedia.org/wiki/Pontos{}_por{}_polegada)
- [48] “irregions @ ipac.caltech.edu.” [Online]. Available: <http://www.ipac.caltech.edu/outreach/Edu/Regions/irregions.html>
- [49] “14103\_pt @ www.ni.com.” [Online]. Available: <http://www.ni.com/white-paper/14103/pt/>
- [50] “Optical\_coherence\_tomography @ en.wikipedia.org.” [Online]. Available: [https://en.wikipedia.org/wiki/Optical{}\\_coherence{}\\_tomography](https://en.wikipedia.org/wiki/Optical{}_coherence{}_tomography)
- [51] “Local @ Pages.Cs.Wisc.Edu.” [Online]. Available: <http://pages.cs.wisc.edu/{~}bolo/shipyard/neural/local.html>
- [52] T. E. D. Campos, R. S. Feris, and R. M. C. Junior, “A Framework for Face Recognition from Video Sequences Using GWN and Eigenfeature Selection,” *In Proceedings of I-WAICV*, pp. 141–145, 2000.
- [53] “intro6 @ radio.feld.cvut.cz.” [Online]. Available: <http://radio.feld.cvut.cz/matlab/toolbox/images/intro6.html>
- [54] “What-algorithm-is-used-to-convert-an-image-from-RGB-to-Grayscale @ www.quora.com.” [Online]. Available: <https://www.quora.com/What-algorithm-is-used-to-convert-an-image-from-RGB-to-Grayscale>
- [55] M. M. Turk and A. A. Pentland, “Face Recognition Using Eigenfaces,” pp. 72 – 86, 1991. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs{}\\_all.jsp?arnumber=139758\\$\delimiter"026E30F\\$http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=139758](http://ieeexplore.ieee.org/xpls/abs{}_all.jsp?arnumber=139758$\delimiter)

- [56] R. Cutler, "Face Recognition Using Infrared Images and Eigenfaces," *Computing*, pp. 1–5, 1996. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.3051&rep=rep1&type=pdf>
- [57] J. Wilder, P. J. Phillips, C. H. Jiang, and S. Wiener, "No Title Comparison of Visible and Infra-red Imagery for Face Recognition," in *Proceedings, International Conference on Automatic Face and Gesture Recognition*. Killington, VT, USA, 1996, pp. 182–187.
- [58] "1092960\_Ford-and-Intel-Use-Facial-Recognition-To-Improve-in-Car-Tech-Safety @ Www.Motorauthority.Com." [Online]. Available: [http://www.motorauthority.com/news/1092960\\_ford-and-intel-use-facial-recognition-to-improve-in-car-tech-safety](http://www.motorauthority.com/news/1092960_ford-and-intel-use-facial-recognition-to-improve-in-car-tech-safety)
- [59] "project-mobii-3 @ www.clubedecriacao.com.br." [Online]. Available: <http://www.clubedecriacao.com.br/ultimas/project-mobii-3/>
- [60] "2016\_G20\_Hangzhou\_summit @ en.wikipedia.org." [Online]. Available: [https://en.wikipedia.org/wiki/2016\\_G20\\_Hangzhou\\_summit](https://en.wikipedia.org/wiki/2016_G20_Hangzhou_summit)
- [61] "chinese\_police\_cars\_scan\_face @ shanghaiist.com." [Online]. Available: [http://shanghaiist.com/2016/04/01/chinese\\_police\\_cars\\_scan\\_face.php](http://shanghaiist.com/2016/04/01/chinese_police_cars_scan_face.php)
- [62] "Harman\_International\_Industries @ en.wikipedia.org." [Online]. Available: [https://en.wikipedia.org/wiki/Harman\\_International\\_Industries](https://en.wikipedia.org/wiki/Harman_International_Industries)
- [63] "content @ www.harmankardon.com." [Online]. Available: <http://www.harmankardon.com/content?ContentID=automotive-mercedes-benz>
- [64] "auto-anti-theft-system-based-on-face-recognition-and-vehicle-telematics @ harman.in." [Online]. Available: <http://harman.in/blog/auto-anti-theft-system-based-on-face-recognition-and-vehicle-telematics/>
- [65] "sdlc\_waterfall\_model @ htm.tutorialspoint.com." [Online]. Available: [https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)
- [66] "valida% C3%A7%C3%A3o-do-sistema @ magnetimarelli.com." [Online]. Available: [https://www.magnetimarelli.com/pt/business\\_areas/powertrain/compet%C3%A7%C3%A3o-do-sistema](https://www.magnetimarelli.com/pt/business_areas/powertrain/compet%C3%A7%C3%A3o-do-sistema)