Visual intention of interaction for HRI using gaze, posture and gestures

Afonso Maria Gonçalves Soares

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Alexandre José Malheiro Bernardino
Prof. José Alberto Rosado dos Santos Victor

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Alexandre José Malheiro Bernardino
Member of the Committee: Prof. Pedro Daniel dos Santos Miraldo

December 2019

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Dedicated to my family and friends...

Acknowledgments

I would like to thank to all the people that helped me through this academic year and that helped me during this master thesis.

First of all, I would like to thank to my thesis coordinator, Professor Alexandre Bernardino who demonstrated great availability and support, helping me a lot in developing the solutions for this thesis and also who gave me the opportunity to work in an exciting environment and with a challenging project.

Also i would like to thank Professor José Santos-Victor for this project opportunity, and I want to thank to all the people in Vislab that were very helpful and nice.

I would like to thank to all the people that integrated the Professor Bernardino team of projects that he was coordinating, João Avelino, Ana Rita Cóias, Diogo Caldas de Oliveira, João Moreira Alves and Igor Lirussi for being a great work partners and helping when I needed.

I would like to thank my parents Isabel Gonçalves and Miguel Soares, to my sister Catarina Soares and my girlfriend Rita Pinto for always supporting me during my academic course and everyday. I would like to thank to all my friends, especially the ones that shared this academic course at my side.

In the end, I would like to thank to Instituto Superior Técnico (IST) for being an incredible good university and for helping me to grow as a professional and as a person.

Resumo

Nesta tese de mestrado é proposta uma ferramenta para detectar a intenção de interação de um humano caso este queira interagir com um robot, utilizando como sensor uma câmara RGB. Esta detecção é dividida em dois módulos sequenciais. O primeiro é um detector de potenciais intenções de interação, que tem como objectivo inferir se uma pessoa está habilitada a interagir com o robot, isto é, se é um potencial indivíduo a vir a interagir com o robot. Este detector é constituído por um sistema multimodal que utiliza a informação de 3 modalidades, a direcção do olhar, a posição da cabeça e a postura do indivíduo, sendo possível no fim inferir a potencial intenção para interagir. O segundo módulo é um detector de gestos dinâmicos que procura analisar os gestos realizados pelo humano para iniciar uma interação com o robot, sendo possível ao robot concluir que gestos foram feitos de forma a realizar uma resposta adequada ao que foi realizado. A arquitectura proposta leva a que os gestos apenas sejam analisados caso o humano possa vir a ter intenção para interagir. Para cada uma das modalidades, para a junção destas e para os processos associados ao detector de gestos, foram estudados classificadores para cada um destes casos, de forma a obter aqueles que levam a uma melhor performance. No fim, a ferramenta foi testada e avaliada para 31 pessoas.

Palavras-chave: intenção de interação, detector de potenciais interações, direcção do olhar, posição da cabeça, postura, detector de gestos dinâmicos

Abstract

This master thesis proposes a tool to detect the intention of interaction of a human if he wants to interact with a robot, using an RGB camera as a sensor. This detection is divided into two sequential modules. The first is a potential intent detector, which aims to infer if a person is able to interact with the robot, that is, if it is a potential individual to interact with the robot. This detector consists of a multimodal system that uses the information of 3 modalities, the eye gaze, the head pose and the posture of the individual, being possible in the end to infer the potential intention to interact. The second module is a dynamic gesture detector that seeks to analyze the gestures performed by the human to initiate an interaction with the robot. The robot can conclude which gestures were made to perform an adequate response to what was performed. The proposed architecture allows gestures to be analyzed only if the human has a potential intention to interact. For each of the modalities, for their final combination and the processes associated with the gestures detector, classifiers were studied for each of these cases, to obtain the ones that lead to better performance. In the end, the whole tool was tested and evaluated to 31 subjects.

Keywords: intention of interaction, potential intent detector, eye gaze, head pose, posture, dynamic gestures detector

Contents

List of Tables

List of Figures

Acronyms

CLNF Conditional Local Neural Fields.

CNN Convolutional Neural Network.

FPS Frames per Second.

HMM Hidden Markov Model.

HRI Human Robot Interaction.

KNN K-Nearest Neighbors.

PAF Part Affinity Fields.

RGB Red Green Blue.

RGB-D Red Green Blue - Depth.

ROS Robot Operating System.

SVM Support Vector Machine.

Chapter 1

Introduction

Robots and their use these days is increasing and is very important in some areas. The impact of this use in areas such as medicine, factory, automation, entertainment, among others, is very present and it is possible to be noticed more and more [1]. The use of robots in various areas makes the Human Robot Interaction (HRI) also an important point of investigation. Areas like psychology, the social and cognitive sciences, artificial intelligence and more, are studied using this topic as the main reference to investigate how a human can communicate and interact with a robot [2].

For the HRI The robot needs to be able to obtain information about the human to interact with him. A communication between two persons can be verbal or nonverbal, just as in a communication between a human and a robot. Ogden and Dautenhahn [3] state that while speech is a primary and key aspect of communication, nonverbal forms of communication have plenty of information for the robot, such as facial expressions and gestures. Using nonverbal communication is possible for the robot to detect the information that the human is transmitting. Breazeal et al. [4] states that, for human-robot communication, the gestures with the head, arms and hands are important but the eyes movement and eye contact are also relevant. Sanghvi et al. [5] states the relevance of posture and body motion in the prediction of interaction between a human (in this case a child) and a robot. Therefore is possible to consider some features of nonverbal communication to have this human-robot interaction, like the eye gaze, posture and gestures.

Based on what has been said above, the main topic that this master's thesis seeks to study is the importance of eye gaze, posture and gestures in the interaction between humans and robots. Therefore the study is based on nonverbal communication and a multimodal approach seeks to divide intention of interaction into two components: the potential intention to interact and the interaction itself. Methods for these two components were studied, tested and in the end evaluated.

1.1 Motivation

Social interaction in robots is increasing and the technologies and methods used in this interaction need to be improved. This improvement can get precise interactions, more humanoid motion for the

robots and even cheaper tools for the various areas where HRI is important and is growing up. These interactions could be beneficial to subjects where the person only wants to interact with the robot in a social way [6], or to teach something to the robot so that is capable of improve is interactions through artificial intelligence [7], or to use this interactions to inform humans, like how a group of persons should flow most optimally and securely in cases where there's a lot of people walking in the same space [8].

Being said this, this improvement can be done through various ways because there's a lot of methods that can be implemented to improve the performance of the interaction with the robot. For detecting the posture of an individual Cao et al. [9] uses a method where receiving as input a color image, produces, as output, the 2D locations of the joints for each person in the image. This is an approach for the robot to get the posture of a person at a certain time. However a different approach was made by [10], receiving also a color image as input, the image is cropped in many images and those small images with an assistance of a machine learning classifier are identified as the limbs of the person. In the end, the output is a set of images that identifies each part of the body of the human and putting all together the posture can be obtained.

Tapus et al. [11] states there exist a large number of signals that can be captured for modeling a person: speech, facial expressions, gaze, gestures. With this, we are capable to obtain the *intention of interaction*. Knowing this, the main problem to study in these areas is to identify and choose the best signals to use to infer what is pretended.

Heinze [12] says the intention of recognition is identified, in general terms, as the process of becoming aware of the intention of another agent and, more technically, as the problem of inferring an agent's intention through its actions and their effects on the environment. We can divide this in two different aspects: potential intent of interaction and the interaction itself. The potential intent of interaction is inferring if the agent is a possible agent to interact and has an interaction posture to the robot. The interaction itself denotes all movements, gestures or speech that are made and intended to send a message allowing communication. This combination gives the intention of interaction.

Identifying the need for interaction is important for the robot. It's still an open problem detecting and knowing what a human is doing to interact with a robot according to [2]. Trick et al. [13] states that a multimodal approach for intention of interaction is beneficial because having a lot of modalities of information for the input, it offers the possibility to compensate modalities that some people don't have, e.g. a person with a speech disorder, and integrating information of several modalities, the uncertainty about the predicted intention can be decreased.

Some modalities can be used to obtain information about the person. Hansen and Ji [14] states that eye gaze is a good tool and it would be important and good to explore the combination of the eye gaze and gestures. Nehaniv et al. [15] claims that understanding human gestures is an important tool for interaction. The posture is also an important modality according to [5]. Using modalities like the eye gaze and posture for the detection could be enough for some HRI. However, Stiefelhagen et al. [16] claims that sometimes the eye gaze can't give good information about the attention of the actors. They say that is necessary to use the information of the head pose together with the eye gaze to obtain good results.

Also about the intention of interaction, Salvado [6], in is Master thesis, states the importance of gestures in the HRI, and evaluates some static gestures in an interaction. As an improvement to his thesis, stated in his future work, I used a dynamic gesture detector for the intention of interaction and studied is performance. Besides that in [6] the focus wasn't the gestures because was used a Microsoft tool, Visual Gesture Builder, and analyzed the performance of this tool. The proposal in this thesis was to analyse a different method used to obtain the information about the gestures.

In nonverbal interactions gestures are the main source of information, and with this in mind, for a person to make a gesture to interact, the person is expected to present an intention to the individual with whom they want to interact, for example by looking at it and with a proper posture. In these cases, the detector for the intention of interaction can be divided in two aspects: a potential intention detector, which analyzes potential individuals that may want to interact, and a gestures detector, which analyzes the gestures made by the individual to start an interaction.

I purposed a multimodal approach using the eye gaze, head pose and posture to detect the potential individuals that want to interact with the robot and a dynamic gesture detector to finalize the detection of the intention of interaction. For each modality was made a study about the best machine learning classifier to use and to infer the importance of combining these three modalities for the intention of interaction.

Given these aspects, the robot must have sensors capable of obtaining this information and to be able to make a decision regarding the human's intention to interact and what kind of interactions he is making. Most robotic instruments nowadays have a built-in Red Green Blue (RGB) camera and a depth camera is also common and popular, so in short anRed Green Blue - Depth (RGB-D) camera, like Kinect. Although the RGB-D sensor is a good tool for gesture recognition because of its ability of skeleton extracting and tracking, its size is bigger than traditional RGB cameras, its price is higher, and its working distance is limited (due to the depth sensor) as stated by [10]. Many robots use only RGB cameras but in RHI a lot of the work and investigation done are focused on RGB-D. Being RGB cameras cheaper than RGB-D and knowing that nowadays almost all instruments that have some interaction with humans have these depthless cameras, it is proposed in this master thesis a tool for HRI that uses only an RGB camera as a sensor for information about the intervening agent.

In this thesis is proposed an improvement in the area of HRI. Is intended to study how and what aspects can be used for social interaction in robots. Besides this, is also intended to create a cheaper tool for this social interaction with the use of only an RGB camera. To test this created tool was used a robot. This robot was Sanbot that is represented in Figure 1.1[1].

1.2 Related work

As explained before, the main problem of this thesis is to construct an intention of interaction detector. A big problem also in this thesis is the use of gestures in this interaction detector. So, for each of them are

[1]This Figure was extracted from https://www.robot-rentals.com/robots-for-rent/sanbot-robot/. This website was accessed at 30/10/2019

Figure 1.1: Sanbot - robot used in this thesis.

presented scientific works that were done and are summarized the problems and challenges associated with each technique.

1.2.1 Related work to the intention of interaction detection

One of the most important features used to infer if the person wants to interact with the robot is the eye gaze. The gaze is associated almost every interaction between two individuals.

Salvado [6] uses the eye gaze and head pose to infer the intention of interaction. He did a study to find out whether using only the eye gaze was sufficient or whether the combination of gaze direction and head position was better. The author was able to conclude that the combination was a better choice and enabled him to obtain better results. By increasing the distance between the robot and the individual, the direction of the eye information is no longer viable due to poor eye movement detection. This way, by combining the head pose with the eye gaze, good results can be obtained even when the human is at a certain distance from the robot. This features used, are obtained with the help of tools explained in [17] and [18].

Gaschler et al. [19] created a tool for a bartender robot, so it could be capable of understanding the persons that want to interact with the robot. Using the head position and posture of the persons as the input information, the robot could infer if the person wanted to interact. In this work, the authors concluded the importance of the head pose for social engagement.

The posture is also a key aspect to know the intention of the human about wanting to interact with the robot. There are many ways to get the posture of a person. In [10] the posture is obtained cropping the input image of a person, while in [9] the 2D locations of person body joints are calculated. How we use this information can be very different. For the first method used, the algorithm is slow and takes some time because of the amount of data image that uses. The accuracy of the posture weren't very

good also. The second method, in another way, has good run time performance and good accuracy.

Nakai et al. [20] uses the 2D locations of the body joints to determine the pose of a person in a shooting position in a basketball game. With this information, the objective is to predict the accuracy of the throw having the player posture. The authors calculated the width of arms, width between the begin of the neck and the person's hands when throwing the basketball, among others.

Lee et al. [21] uses the 2D joint locations of the arms and torso and calculates the angles that each joint does with another joint, being the objective of this work, to help the user learning tai-chi poses. With this information, the authors create a classifier that gives the user information about is actual tai-chi posture. With this, the user is capable of knowing if is doing the right posture according to the exercise.

Concluding, the use of the eye gaze should be combined with the head position because is hard to detect eye movement at certain distances. Besides this, the posture could be obtained throw many ways, however getting the 2D or 3D locations of the joints allows using that information in many ways for classifying the posture of the person, for example using distances or angles between joints (both works studied showed good accuracy and performance). This choice is made accordingly to the problem in question.

1.2.2 Related work to gesture detection

To gesture detection, there's a lot of methods that can classify which gestures are most likely to being made. However for static gestures cropping the image and classifying the cropped image is more efficient while for dynamic gestures, the use of Hidden Markov Model (HMM) is more efficient because if we are capable of segmenting the dynamic gesture, the HMM gives the probability of a certain sequence being a certain gesture. The use of a HMM in a problem like this, a dynamic gestures problem, is analyzed and exemplified in [22]. In this paper the authors use this method for the robot to analyze sign language, this is, the sequence of symbols done with the hands of the individual interacting with the robot, obtaining good accuracy in the realized tests. Although, the authors use segmentation of the sequence of symbols image and cropping to obtain images of the hands and classifying each of them to which symbol are being made. So for the use of HMM, in this case, it also needs a good feature extraction.

Liu and Wang [23] use as feature extraction the motion, the optical flow. Based in the optical flow is possible to obtain a Histogram of Oriented Objects (HOG) which divides the image into blocks and for each image a histogram is calculated. The problem with this approach is being based in optical flow, because it only has good performance when detects only one moving person in the image, so the background needs to be always static. Qiao et al. [24] use a tool according to [9], where they can obtain the 2D or 3D locations of the body joints of the human presented in the input image. With these locations distances between the wanted posture and the actual posture are calculated and a score is given according to similar or different postures. This paper demonstrates the use of the points given by a tool that gives the location of human joints. As said before, in the related work to intention of interaction detection, the information of this joints can be used in many ways, like calculating the angles of each

joint does with another joint, being possible for the robot to classify a gesture Lee et al. [21].

These solutions take into account several methods that could be used to segment the dynamic gestures, to feature extraction and to classify each gesture. According to the problem in question, only the junction of some of these methods could give a solution.

In conclusion the use of a tool like Openpose [9] to obtain the the 2D locations of the human joints, the segmentation of the movement when the human is performing a gesture, the classification of each segment according to the angles of the joints and the probability for each gesture given a determined sequence that all together describe a gesture, is a good way to get the information for the gestures detection.

1.3 Objectives

The main objectives in this master thesis are the study of a intention of interaction detector, combining a potential intent detector and a gestures detector capable of knowing if the person wants to interact and which gestures were made by the person that is interacting with the robot.

In the first main objective, it was intended to create a multimodal system capable of deciding if the person is a potential agent to interact. Was pretended that the used modalities are the eye gaze, the head pose and the posture. For each of theme was intended to develop some classifiers and study the performance of the classifiers used in each modality and choosing the classifiers that had the best results. Having a classification for each modality, this is, a value indicating if the person is looking to the robot or not, if the head position is adequate or not and if the posture is a interaction posture or not. For the classifiers was necessary to create datasets for each modalitie and train and test them.

For the second main objective, it was intended to develop a system capable of detecting the gestures made by the user. This system should be capable of identifying some gestures that were indicated during the development of the constructed classifier for the gestures. Also was needed to create datasets of the gestures indicated to train and test the classifier.

In the end, some experiments were pretended to evaluate the precision of the detector of intention of interaction and the gestures classifier.

1.4 Thesis Outline

In the following chapter, “Background” (Second Chapter) , an overview of the general theoretical concepts related to this work is given. After the overview each concept is discussed in detail. Those concepts are the multimodal approach, the detector of intention of interaction, in which fit the eye gaze classifier, the head pose classifier and the posture classifier, and in the end, the interaction classifier,this is, the gestures classifier.

In the Third Chapter, there is the Methodologies. In this chapter is explained the eye gaze and head pose module, the posture module, the intent of interaction module and the gesture module used in this thesis.

In the Fourth Chapter, the Implementation. Here it is explained the robot sensors, the communication processes and the state machine used to control the robot in the implemented tests.

In Fifth Chapter, there's the Results and Experiments. In this part of the thesis are the description and results of the experiments performed to achieve the objectives proposed.

The last Chapter is the conclusions of this work and future work to do.

Chapter 2

Background

After outlining in chapter 1 the aspects to study and to deepen, it was possible to divide the work to be done, an intention of interaction detector, in two points. The first refers to the potential intent detector and the models used to obtain the information and classify it so that the robot can infer if the person is a potential agent to interact with it. The second point refers to the gesture detector and the models used for the robot to identify the gesture that the actor performs so that it can interact with that actor correctly, as would be expected in social interaction. So this chapter presents the bases and studies used in problems close to or identical to those intended to be studied.

First is presented what is intended for a potential intent detector and what models and tools could be used to improve performance and achieve satisfactory results in relation to the intended. In this detector, some approaches already performed to the 3 modules (eye gaze, head pose, and posture) are mentioned individually.

Finally, it is also presented what is thought for a gesture detector, in this case, dynamic gestures, and some approaches already studied that can be used.

2.1 Intention of interaction detector

In this section we explain what is the intention of interaction as well as some works and studies carried out, considering this aspect. This section also refers to the intended multimodal approach by using more than one modality of information so that the robot can infer the intention.

2.1.1 What is the intention of interaction?

As said in Chapter 1, Heinze [12] describes the intent of interaction as the process of becoming aware of the intention of another agent. In a social engagement, is important for the human to become aware that exists an interaction intent, through some signals or information obtained from someone or the environment. When a person wants to interact with someone, needs to send some signal to who wants to communicate. Whether through gestures, speech or other means, such as using the environment around them to send a message, it is always crucial that an actor wishing to initiate a social interaction

be able to get the other actor, or actors, to engage your attention so that communication is received and interpreted correctly.

Benkaouar and Vaufreydaz [25] say that recognition of intentions is an subconscious cognitive process vital to human communication, something that our brain can conclude based on certain information in an intuitive way. This means that this ability is something that humans acquire and learn early in life, especially when they start interacting with other humans. The authors also say that the most important and relevant signals are the non-verbal signals. This statement makes it possible to deepen the study of intention in nonverbal cues, such as one's posture and gaze.

Although the information acquired through the intervening is the most important, the surroundings should also be considered as a very important aspect in the intention of interaction. Kim et al. [26] argues that robots are capable of inferring human intention through the relationship between the agent and the environment. For this, the authors argue that a model capable of leading the robot to learn the interactions that the human has with the environment, make this detection of interaction intent possible. This idea is based on how the human being acquires this ability because it acquires by interacting with others and observing the surroundings, that is, other interactions that, even if not intended for them, allow us to learn what are the signs that indicate this intention.

Robots need to have this ability that humans have. Having other skills that may be important to infer interaction is something crucial so that they are capable to have this ability to. Robots have a variety of sensors capable of performing various actions, such as tracking movements, identifying certain actions or positions, detecting the location of certain specific points, among others, and using all this information to infer something correctly and extremely quickly due to its computational power.

Several authors report various modalities and studied them for acquiring information for the detector of intention of interaction. Many of these authors use gestures as one of the inferences modality [6], however in this thesis for the intention detector the gestures are used in a sequential way and not in a parallel way, using this information only when the robot detects a potential agent that intends to interact with the robot. Therefore, for the robot to be able to detect if the human has a possible intention to interact, other modalities are used, and gestures, only when the robot detects that. The gestures also are used to understand what kind of interaction the human is doing so that the robot can give an answer.

Concluding, is used a multimodal approach to be obtained information about the person for the potential intent detector and the gestures to understand what the person is trying to communicate the robot.

2.1.2 Potential intent detector

The potential intent detector denotes all the modules needed to make it possible to infer if a person is in a position to have an intention to interact. Consider a presenting robot, it presents aspects to a person and that person is looking at the robot and is focused on the robot, however that person may not want to interact with the robot. To interact with the robot, although the individual is already paying attention to the robot, they only need to perform some action to interact with the robot. It is based on this idea that

the intention of interaction detector wants to be divided, because the robot first defines potential agents to interact and only later, in the case of nonverbal interactions, through gestures, the robot detects that there is an intention to interact.

Multimodal approach

Most of the time, the use of more information is beneficial. A multimodal approach is the use of various modalities of information to make a final decision. In the case of HRI, a multimodal approach indicates that aspects such as speech, posture, gaze direction, movement, environment, and others are used. Each of these modalities represents a set of information and with this information, classifications are made in each of the modalities. These classifications indicate for each modality the level of interaction intent or even what type of interaction is involved. Having these results for each of the modalities, it is intended to make a final decision as to the intent of interaction or interaction, and it could be given the same or different costs for each outcome of the modalities. Figure 2.1 presents this logic for 4 different modalities (speech, gestures, gaze direction and scene objects), and an assistive robot seeks to give an elderly person the object he wants to grab.

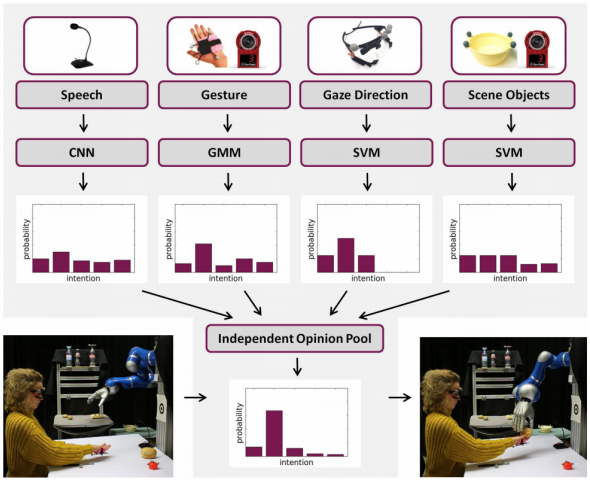


Figure 2.1: Multimodal approach scheme adapted from [13]

Trick et al. [13] studies the benefits of a multimodal approach. The author claims that the use of more than one modality is beneficial because some modalities can compensate lack of information or malfunction in other modalities and using more information that is obtained from different ways, the accuracy of the correct final decision is greater. In this work the principal point was to reduce the uncertainty of the prediction of intention, using a combination of various modalities, in which case the ones explicit in Figure 2.1. The authors concluded that the uncertainty can be decreased using multiple modalities and even inaccurate classifiers can contribute to this reduction or to improve performance and robustness when using many modalities.

Knowing that increasing modalities improves performance and decreases the uncertainty, 3 modalities have been taken for the potential intent detector to study and build. As mentioned earlier, using an individual's gaze direction, head position, and posture are good informational modalities for an HRI

detector [6] and [20].

Benkaouar and Vaufreydaz [25] also use a multimodal approach and studies which modalities are more relevant for an engagement detection in one-user and multi-user scenarios. The authors concluded that the most relevant modalities in the feature set were: shoulder rotation, face position and size, user distance and lateral speed and sound localization. These conclusions are in accordance with the modalities chosen for this work, as facial features and posture features are important.

The modalities that this thesis wanted to study are the eye gaze, the head pose, and the posture. For each one of them, is presented the backgrounds used for the creation of the modules.

Eye gaze module

Eye gaze is one of the most important modalities in inferring if a person wants to interact with someone. In most social interactions, the eye gaze is crucial because it indicates who we want to address and if an individual wants to initiate an interaction, he or she usually directs his or her gaze to the person, place, or object that they want to interact with.

There are several studies regarding different methods to obtain the direction of an individual's gaze. Salvado [6] did a long study of several methods, identifying some that need higher computing power than others due to the use of Machine Learning in their implementations.

One of those works that don't use machine learning, Timm and Barth [27], has is basis in the importance of localizing the center of the eyes for problems in the domain of computer vision applications such as face recognition or eye-tracking. The authors refer in is work an approach for the creation of an algorithm based in image gradients. They computed the squared dot product between the displacement vector of a center candidate and the image gradient for every pixel, and with this, the position of the maximum corresponds to the position where most image gradients intersect. They concluded that the created algorithm was a low computational complexity and a very high accuracy especially for special scenarios such as pupil localization, compared with other state-of-the-art methods.

One of the works that have machine learning fundamentals, has as approach the creation of a tool named Openface [17]. In the OpenFace solution, there is an input of an image or a sequence of images. After that the face is detected with dlib tool [28], and after that is used Conditional Local Neural Fields (CLNF) [29] for facial landmark detection and tracking. Next, they do eye-gaze detection by first detecting the location of the eye and the pupil, using their CLNF model. Having those locations, that information is used for computing the eye gaze vector for each eye. They also do a head pose estimation. In Figure 2.2 is demonstrated the Openface pipeline.

As said by Salvado [6], many of the solutions that he found aren't open source and not easy to replicate. Although OpenFace is an open-source tool with the code needed and the models. OpenFace can give the gaze of the two eyes, the head position and the facial landmarks, so is a good tool to use to obtain the eye gaze which is the wanted in this thesis. In Figure 2.3 is demonstrated the use of Openface in four different situations, being two of them a demonstration of the head position and gaze at long distances and the two others a demonstration of this features at close distances.

Having [6] approach as a good basis in the aspect of eye gaze, and knowing that the author had

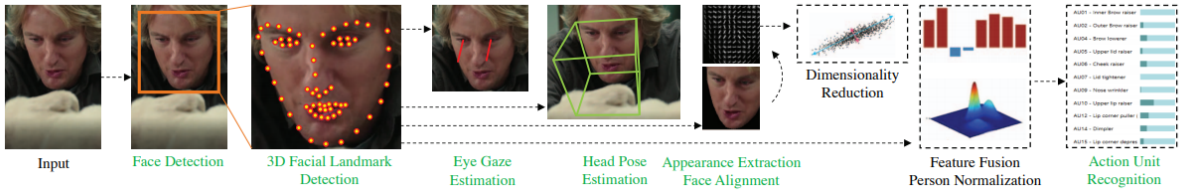
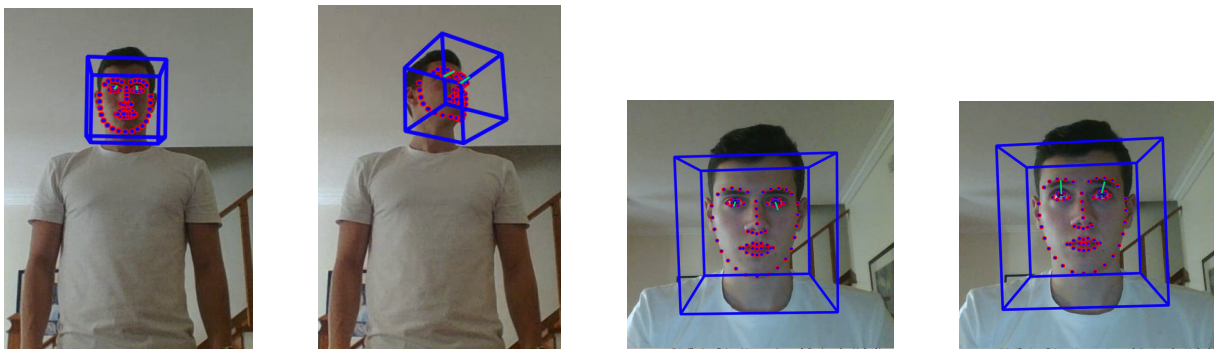


Figure 2.2: Pipeline of OpenFace extrated from [17].



(a) Person looking ahead at long a distance.

(b) Person looking up at a long distance.

(c) Person looking ahead at close a distance.

(d) Person looking up at close a distance.

Figure 2.3: Demonstration of the Openface in a person for long, (a) and (b), and close, (c) and (d), distances.

good results with his approach, in this thesis is tested a different classifier for the eye gaze. The Openface gives the gaze vectors for the two eyes and these vectors point to the direction where the eyes are looking. With these vectors that are normalized, the Openface calculates the angle in each 3D coordinate that each eye vector does with the optic center of the camera. Having this information the main feature for eye gaze are those angles. So having this in mind, some values give an eye gaze directing to the robot or directing to other points than the robot. A classifier that is possible to use is based on Bayesian risk minimization [30]. A Bayesian risk minimization refers to a decision theory that is informed by Bayesian probability. It is a statistical system that tries to quantify the tradeoff between various decisions, making use of probabilities and costs, minimizing these. So having probabilities of a person looking or not looking to the robot, according to some feature values, this approach will find a boundary for those values, so it is possible to classify whether a person is looking at the robot or not.

Head pose module

Using the OpenFace tool, explained in the section above, it is possible to obtain the head position also. Why use the head pose? First, a person may be looking at the robot but has an unsuitable head position, i.e. someone who is looking sideways at the robot, and sometimes this type of head position does not indicate that the person wants to interact with the robot even if you are looking at it. Second, Salvado [6] states that the OpenFace doesn't work well at long distances for the eye gaze. At a certain distance, the tool isn't capable to detect where the eyes are targeted, so the head pose can give where the face of the person is targeted and consider that new gaze as the eye gaze. The author also concludes that

combining eye gaze and head pose gives better results than using just the eye gaze or the head position.

This tool gives the angles of the head according to the center of the camera, or in other words, OpenFace puts a 3D box around the head of the person and detects the face. The head pose detector of OpenFace returns the person's head pose in the camera reference frame as an axis-angle representation. In Figure 2.3 is possible to visualize the box mentioned before.

For the head pose module two classifiers were studied. The K-Nearest Neighbors (KNN) [31] and the Support Vector Machine (SVM) [32], because are fast classifiers and demonstrated a good performance in [6].

K-Nearest Neighbors is a machine learning algorithm that can be used for classification problems with supervised learning where the labels are given, or it can be unsupervised learning where the labels are not given. This algorithm uses the k nearest neighbors to classify the data into a known class. Let us suppose a set of data labeled by two colors (green and red) described in Figure 2.4. If we want to know which class represents the new black dot represented in the last Figure, using the k-nearest neighbors, we need to choose the number of neighbors that we want to use. If we use 3 neighbors like in the Figure, the algorithm will search the closest data of the point that we want to classify, and when the algorithm finds 3 points of data of the same label the new unlabeled point is labeled the same class as those 3 points. In the example, the black dot is labeled as a green class using 3 neighbors. The same logic is used to another number of neighbors.

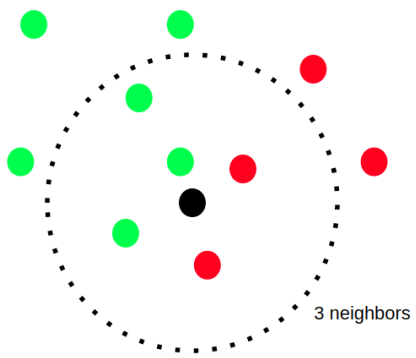


Figure 2.4: Example of k-nearest neighbors.

Support vector machines is used typically as a supervised algorithm in machine learning for classification problems, the objective of the support vector machines is to create a hyperplane or a set of hyperplanes that is capable of separating the data. Cortes and Vapnik [33] describe this hyperplane's boundary by the following equation:

$$w.x + b = 0 \tag{2.1}$$

The vector w is normal to the hyperplane and is the perpendicular distance from the hyperplane to the origin. The objective of the SVM is to create a hyperplane that is closest to the data of both classes. This can be seen in Figure 2.5.

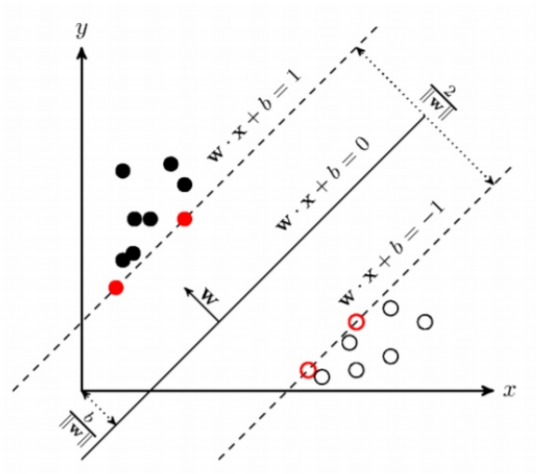


Figure 2.5: 2D hyperplane through two linearly separable classes, extracted from [6].

The points closer to the hyperplane are called support vectors, like described in the Figure above, being those points the points used to calculate the hyperplane. The hyperplane is equally distant to both margins, and to find the best hyperplane, we need to maximize this distance. The distance is given by $\frac{1}{||w||}$, so when we try to maximize it, is the same that minimize $||w||$. Minimizing $||w||$ is equivalent to minimize $\frac{1}{2}||w||$ so the problem is given by:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(x_i.w + b) - 1 \geq 0 \; \forall_i. \end{aligned} \tag{2.2}$$

To respect to equation 2.3 y_i = +1 in the positive class points and y_i = - 1 in the negative class points.

To handle not linearly separable data, the constraints can be relaxed from the equation above. This can be achieved by introducing a positive slack variable ξ_i , i = 1, ..., L. L is the number of points in the training data, and the problem now assumes the following equation:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{L}\xi_i \\ \text{subject to} \quad & y_i(x_i.w + b) - 1 \geq 0 \; \forall_i. \end{aligned} \tag{2.3}$$

In the equation above, the parameter C controls the trade-off between the slack variable penalty and the size of the margin.

In the end, with the w and b calculated, the SVM classifies the new data as in the ”positive” class if the result is greater than zero or as in ”negative” class if is less than zero.

Posture module

In addition to looking, posture is also a very important aspect of modalities used to let a robot know if anyone wants to interact with it. Cao et al. [9] states that estimating the pose in 2D, in images and videos, is a key aspect for a machine to be able to understand what an individual or a set of them is performing or intends to accomplish. Lee et al. [34] has developed work to create a system that can help people with

stroke rehabilitation. The author states that this system aims to monitor and evaluate user performance in four different rehabilitation exercises. To analyze what the user is doing, the author obtains this information through the individual's posture, showing the importance of the information coming from the posture.

So how do we get the posture? Several authors refer different ways to do this, Gao et al. [10], uses the image of an RGB camera, and with this search seeks the identification of a person when looking for a face of a person. Once a person is identified in the image, an image cropping algorithm cuts the image into several smaller ones and each of these images is analyzed and classified as a body part or not. At the end of this process, the various body parts of an individual are identified thereby obtaining the location of each of these in the image. However, the author concludes that the process is very slow and not very accurate.

Cao et al. [9] created a tool called Openpose. This tool aims to locate the position of the joints of each member of a human being, being able to obtain the posture of one or several people. For this, the tool receives a input image for a Convolutional Neural Network (CNN) [35] to predict the joints. After that is obtained confidence maps for body part detection, i.e. like a probability of being a certain body part. Part Affinity Fields (PAF) for part association are calculated and then the parsing step performs a set of bipartite matchings to associate body part candidates. In the end, the tool assembles all these body joints detected into full-body poses for all people in the image. The pipeline of this tool, explained before, is represented in Figure 2.6.

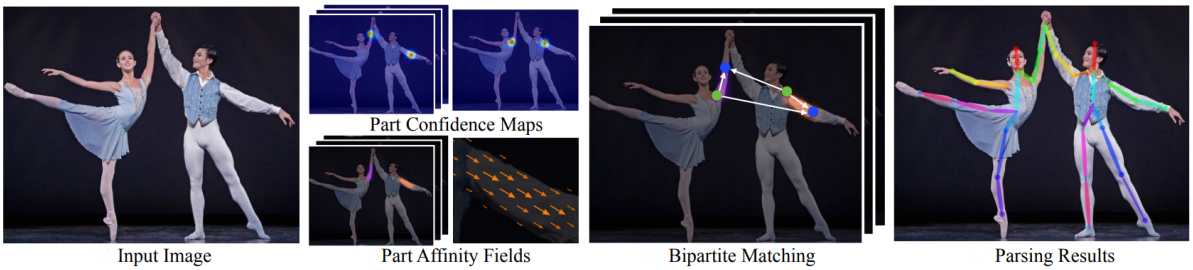
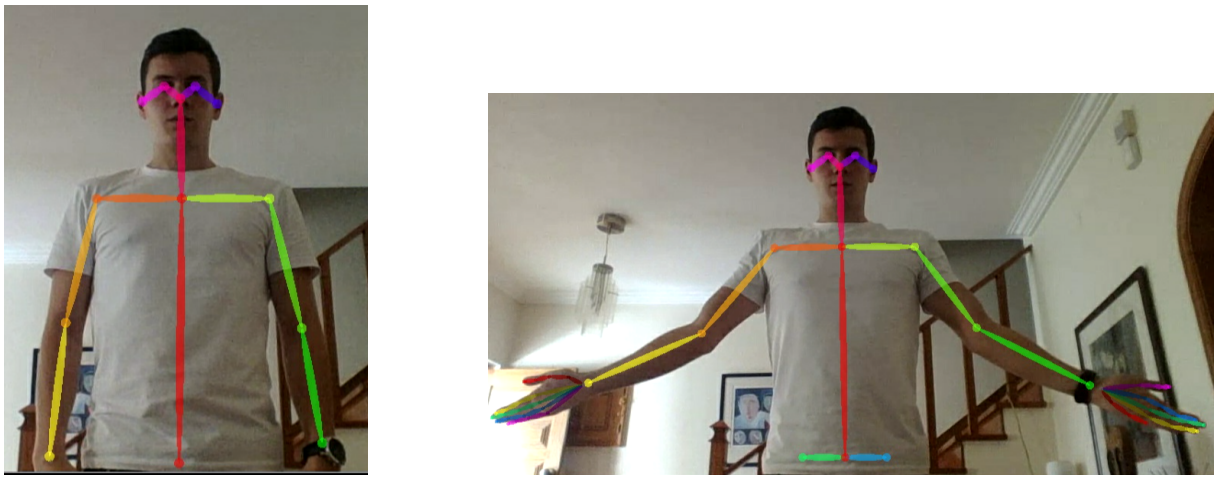


Figure 2.6: Pipeline of Openpose adapted from [9].

Like Openface, Openpose is an open-source tool, with code and models easy to access and use. That said, to remove posture information from the image, this tool was used in this thesis. In Figure 2.7 is demonstrated the use of Openpose in two different body positions.

Having the posture through Openpose, what can we do with this information? Nakai et al. [20] in their work aims to provide a prediction for a free throw in a basketball game, i.e. whether the player makes a position that leads to a hit in the basket or not. The authors use the points of joints given by Openpose and calculate the distances between them and between surrounding areas such as the ground. The results of this work were satisfactory and obtained good accuracy according to the authors. Based on this work it is possible to state that through this tool it is possible to calculate limb lengths or even distances between a certain limb and another zone of the image.

Lee et al. [21], in turn, uses a different approach in her work. Although, he doesn't use Openpose to remove posture information from an image, but uses a Kinect and the Microsoft Developer Network,



(a) Pose with arms closed.

(b) Pose with arms open.

Figure 2.7: Demonstration of the Openpose in a person.

Kinect for Windows SDK. This gives the posture of an individual in the image just like Openpose, so conclusions can be drawn as to how the authors treated the information obtained by the tool they used. The authors' work aimed at a rehabilitation system to perform tai-chi exercises. For this work, the joint information was used to obtain the angles between the human members, in this case allowing an evaluation of the user's movements and indicating if he is putting the members in a correct position and shape. Although this work was just a prototype, the authors conclude that the results were encouraging so that the solution is good.

In conclusion, the use of the Openpose information to calculate distances or angles between joints is a good approach.

Like the head pose module, for the posture module, also two classifiers were studied. The K-Nearest Neighbors (KNN) and the Support Vector Machine (SVM).

2.1.3 Gestures detector

After the detection of a potential intention by the robot, this one needs to be able to detect if a person intends to interact and what the human is doing so that the robot could give some feedback like in a social interaction between two or more persons. For the robot be able to do this, needs a gestures detector, so it can detect the interaction that the human are trying to make. Having a gestures detector is needed to study the concepts of gesture and how to detect and classify a gesture. This section refers to those points.

What is a gesture?

Creider [36] defines the word "gesture" as a set of communicative movements. The author divided gesture into three phases: Preparation (optional), the initial phase; Stroke (obligatory), that is the phase with meaning and effort; Retraction (optional), the final phase. This idea expressed by the author goes according to what most people think when trying to define the word gesture.

Drewes and Schmidt [37] defined gesture has a sequence of elements, like strokes, performed in sequential time order. Mitra and Acharya [38] states three types of gestures: hand and arm gestures, head and face gestures and body gestures. The authors also state that exist static gestures and dynamic gestures. The first ones are less complex because could be obtained only with a simple image, however, the dynamic gestures need a sequence of images or a video to get the movement and classify the gesture.

In this thesis, the gestures analyzed are gestures to interact with the robot and expecting feedback from it. So as said before in Chapter 1, gestures have importance in the detection of intent to interact. The gestures can be used as a modality in the intention of interaction Dautenhahn et al. [39], and using the gestures in this way to infer the intent, is important to define types of gestures that can be grouped so that could be easy for the robot to identify and classify the gestures. Dautenhahn et al. [39] studied this aspect and divided the gestures into 5 groups:

- ”Irrelevant”/Manipulative Gestures. These gestures don’t have intention when they are done. They are irrelevant, which can be side effects of motor behavior, and actions on objects. These gestures aren’t used to communicate

- Side Effect of Expressive Behavior. These gestures represent an expressive way in the communication done between the intervenes. These gestures are associated with humans because it’s something unconscious like moving the hands when talking with someone.

- Symbolic Gestures. A symbolic gesture as an idea associated. These gestures represent symbols that have some mining to the intervenes. These types of gestures are used to substitute some types of communication, like speech. This substitution is made accordingly a code that the person should know. This codes can variate with culture or education. An example of this gesture is the gesture to order the bill in a restaurant.

- Interactional Gestures. These are the most important gestures for the intention of interaction. These types of gestures are used to regulate interaction with a partner, they are used to initiate, maintain, invite, synchronize, organize or terminate a particular interactive, cooperative behavior of interaction. An example of a gesture of this type is a handshake.

- Referential/Pointing Gestures. These gestures refer or indicate objects to the other agents in the environment(humans or robots). Pointing locations in space is an example of this type.

So the gestures are a crucial part of an interaction between a human and a robot, being very important that the robot could understand and identify correctly the gestures, having good interaction with the human (good feedback of the robot).

In this thesis, the gestures are used to detect the intention of interaction and to interact. Also, the gestures used and studied were dynamic, being the gestures module of this work a dynamic gestures module.

Gestures module

The robot, by identifying that an individual is able to interact with, begins to analyze his possible interaction, that is, the gesture he is performing, so that he can respond to that gesture allowing communication between humans and robots. This module aims to analyze dynamic gestures so it will be works that focus on these gestures used as a basis.

The first aspect to study is how to get the information as it is present not only in an image but in a sequence of them. Wilkowski [40] extracts the necessary information using color-based image segmentation methods and introduces high dimensional feature vectors to describe the handshape of in the picture with accuracy. Knowing a-priori knowledge on the construction of the gestures the author can do a dimensional reduction. With this information the author could obtain the hand shape along the image sequences and classify the hand gestures, having the system demonstrated a reliable recognition properties for static and dynamic gestures. Sagayam and Hemanth [22] also use a similar approach for feature extraction, being them work grounded in hand gesture recognition. The authors extract the discriminant feature vector points from the hand gestures using a technique and database described in [41].

Qiao et al. [24] in contrast, use another approach to extract information from movements associated with gestures. The authors use the Openpose tool to obtain the 2D location of human body joints. In this work, the objective was to present the real-time 2D human gesture grading system for monocular images based on this tool. The authors acquired the sequence of images that represent a certain movement and combined the points of the joints located in each image, allowing to obtain a trajectory as shown in the Figure 2.8.

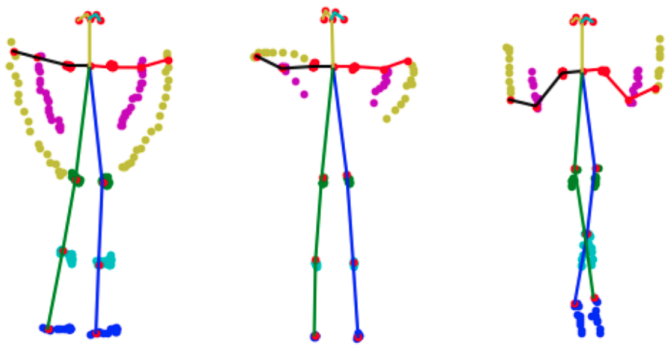


Figure 2.8: Trajectory of joints. Extracted from [24].

For the system to be robust to input noise, one approach is referred to in [24]. The authors redefine the path equations for Bézier curves, that is, these curves are parametric curves consisting of a start point, an endpoint, and multiple control points [42]. This allows these trajectories to become smoother and easier to analyze. Since Openpose is a tool used for other aspects of this thesis and knowing that the results obtained were satisfactory, this work will serve as a great starting point for the gesture detector to be created.

Having the trajectories of gestures is necessary to move to the classification of movements. As

already mentioned, a gesture can be considered as a set of movements. In Wilkowski [40] and Sagayam and Hemanth [22], gestures are segmented and each of these segmentation is classified. That said, it is possible to state that with the total trajectory of the gesture, segmentation results in a sequence of movements, and it is necessary to classify each one of them. Sagayam and Hemanth [22] classifies this data through an SVM, and this algorithm will also be studied in this thesis as classifier of each segment of a given gesture.

Sagayam and Hemanth [22] and Wilkowski [40] refer an approach to identify the gesture after being segmented and classified. As mentioned above the gesture can be defined as a sequence of movements that have a certain order and one movement depends on the previous one. Based on this idea, the authors use the Hidden Markov Models (HMMs) to, based on the classifications obtained and a given sequence, infer what type of gesture is most likely to be performed according to this information.

Sagayam and Hemanth [22] state that a Hidden Markov Model is a stochastic process, which observes the emissions from state transitions of the extracted gesture features. HMM are used to predict a sequence of state changes based on the observed state sequences. The transitions among each state in an HMM are governed by a set of probabilities named transition probabilities. The authors also claim that an HMM-based hand gesture modeling, that can be used also to arm gesture, has three steps:

- The system has finite internal states that generate a set of external events.
- The internal state changes are hidden to a viewer outside the system.
- The current state always depends on the previous state.

In conclusion, these approaches used by all those authors can be combined and create the gestures module pretended in this thesis. A pipeline of a general idea for the gestures module, according to the concepts explained in this section can be visualized in Figure 2.9. This pipeline is the basis for the detector of gestures constructed in this thesis.

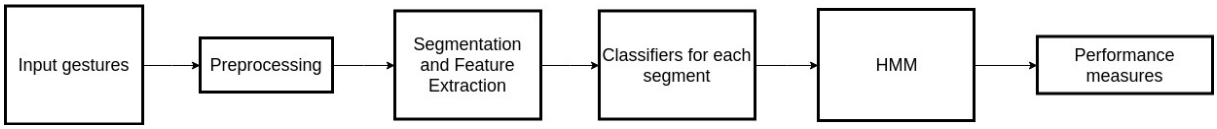


Figure 2.9: Pipeline of the gestures module. Adapted from [22].

Chapter 3

Methodologies

This chapter explains and demonstrates the methodologies that have been implemented in this work, to achieve the desired objectives. The methodologies are divided into a small section to explain basic concepts related to rotation matrices and angle representation, in a section that demonstrates the general architecture of the system that is intended to build deepen each module of that architecture explaining the methods used and assumptions made. The modules constructed are the potential intent module, constituted by the eye gaze, head pose, posture and final intent detector modules, and the gestures module.

3.1 Baselines

Some concepts had to be considered, and the baselines section explains what concepts and why they had to be considered. For the eye gaze and the head position are used angles in a certain representation and referential, so it is important to study and understand these aspects that are important to this master's thesis.

A 3D rotation can be represented by a vector in which each coordinate represent an angle, the rotation angle in each coordinate. These angles are called Euler angles. It's important to understand what Euler angles represent and how to get and apply a rotation matrix because they are crucial aspects for the eye gaze and head pose methodologies.

Also very important is the referential. The referential used for the camera can be seen in Figure 3.1, being the z-axis perpendicular to the center of the camera and pointing to the opposite direction of the center of the camera.

A vector that represents a rotation can be defined by 3.1. Each coordinate represents the Euler angle for each coordinate. With these angles is possible to use axis–angle representation which is equivalent to $\theta = \phi$e. Where ϕ is the angle of rotation and the e is the unit-length axis of rotation, the θ represents the rotation vector. For example, if there is a rotation of π in the x-axis then the rotation vector would be equal to the equation 3.2.

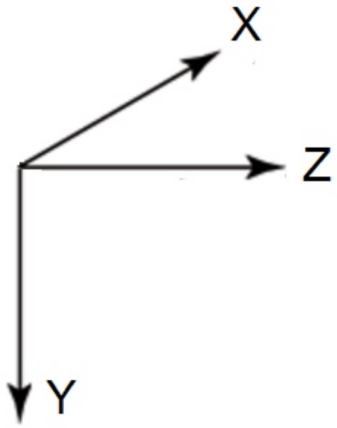


Figure 3.1: Camera referential.

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \tag{3.1}$$

$$\theta = \phi e = \phi \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} = \pi \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \pi \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha = \pi \\ \beta = 0 \\ \gamma = 0 \end{bmatrix} \tag{3.2}$$

Having the Euler angles, the rotation is made through a 3D matrix. Next, are presented the rotation matrices for each axis. A rotation matrix, R, is given by the product of each of these matrices, being the product order dependent on the sequence of rotations that are wanted.

Rotation on x-axis:

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{3.3}$$

Rotation on y-axis:

$$R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{3.4}$$

Rotation on z-axis:

$$R_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Having a vector of Euler angles, is possible to obtain the rotation matrix associated to these angles. The next equation performs that:

$$R = \cos\phi \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + (1 - \cos\phi) \begin{bmatrix} e_x e_x & e_x e_y & e_x e_z \\ e_x e_y & e_y e_y & e_y e_z \\ e_x e_z & e_y e_z & e_z e_z \end{bmatrix} + \sin\phi \begin{bmatrix} 0 - e_z & e_y & \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \tag{3.6}$$

Zelinsky [43] states that is also possible to obtain the Euler angles having a rotation matrix R. For this consider two variables (ϕ, e), where ϕ is the angle of rotation in (0, π), e is the unit-length axis of rotation and t = $R_{11} + R_{22} + R_{33}$ is the trace of R. The rotation vector θ can be calculated according to:

- If t = 3 then R = I, and exists an infinite number of solutions defined by ϕ = 0 and arbitrary axis of rotation e
- If -1 $<$ t $<$ 3, exists a unique solution. Since:

$$\sin\phi \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} = \frac{R - R^T}{2} \tag{3.7}$$

  Then the solution is given by:

$$\theta = \frac{\phi}{2\sin\phi} R - R^T \tag{3.8}$$

  Where $\phi = \arccos\left(\frac{t-1}{2}\right)$.

- If t ≥ -1, exists two solutions defined by $\phi = \pi$ and e_1 = $-e_2$ such that $ee^T \geq \frac{R+I}{2}$.

3.2 System Architecture

In Figure 3.2 is represented the architecture for the system built and that is based on the methodologies that are used in this thesis. So this pipeline presents the architecture for the methodologies, having more pieces of this work that are not mentioned in this chapter. These pieces are the robot sensors, the communication process and the control behavior of the robot, being described in the Implementation chapter.

This architecture was chosen considering that the following is intended:

1. A human approaches the robot.
2. The robot tries to infer if the human wants to interact with him,i.e if the human is looking at the robot and with a correct posture.
3. If the human wants to interact, he/she makes a gesture.
4. The robot, after concluding that the human intends to interact, analyzes the gesture made. Once analyzed, the robot responds to this gesture in the right way leading to social interaction between the two.

The methodologies architecture is divided into the two most important aspects: the potential intent detector and the gestures detector. In the first are discussed the three modules, eye gaze, head pose, and posture. For each of them is discussed how the robot can obtain the information and the classification process for that information, so the robot can infer the final intent in this module, which is also discussed. At last, is also explained the detection of gestures and the segmentation and classification of them in the gestures detector module.

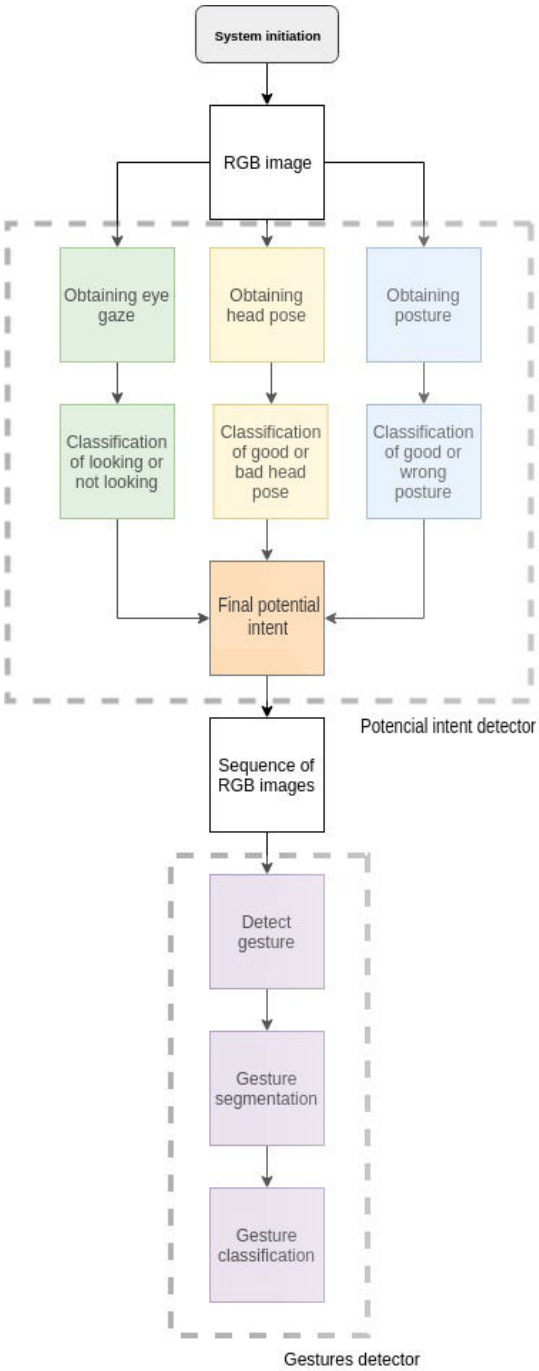


Figure 3.2: System architecture for the methodologies used in the system.

3.3 Potential intent detector

As explained above, the potential intent module seeks to detect whether a human has a possible intention to interact with the robot or not. For this analysis, as already discussed, a multimodal approach is used, and the modalities are the eye gaze, the head position and the posture of the person. With the information of these three modalities and their respective classifications, the robot can decide as to the intention of humans, allowing the consequent analysis of gestures that can be made next.

In conclusion, this module specifies the methodologies used in the three modalities and in the final decider that uses the information acquired from them.

3.3.1 Head pose module

The position of the head is an important modality because at long distances can indicate where a person is looking,i.e in the distance is not so noticeable the movement of the eyes but the head is.

Salvado [6] has had good results using the Openface [17][18][44][45][46] analysis tool and certain methodologies, which makes encouraging the use of this tool and some methodologies applied by the author.

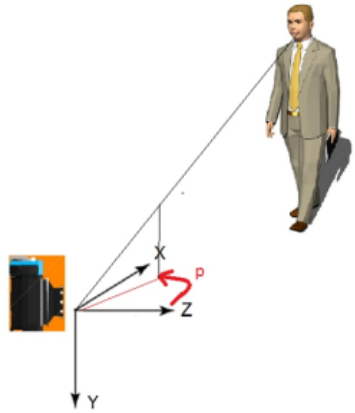
Under this master's thesis, the use of head position is used as an aid in the eye gaze module and as an individual modality. It is used as an aid to eye gaze, as Openface can't correctly detect eye gaze vectors from certain distances (it has been found that it does not detect correctly for distances of more than about 1 meter). At long distances, the eye gaze points to the direction of the face. Therefore the use of the head position allows for long distances to use the direction of the human face as the direction of look, taking into account this aspect because a person tends to direct the face to look, not forcing so much eye movement. In addition to this utility, the head position is used as an individual modality. It is important to note that in cases where Openface can correctly detect the eye gaze vectors because the person is at a favorable distance, then the eye gaze module is mostly defined by these vectors, and a modality that indicates the position of the head is important in those cases. In addition to this, the individual modality can make it possible to correct some malfunction with the gaze direction,i.e the eye gaze module combines the head position information with the look direction, which indicates that if there is a major failure in the direction of the eyes, the position of the head on this module may have no influence, not allowing good information from that module.

Head pose detection

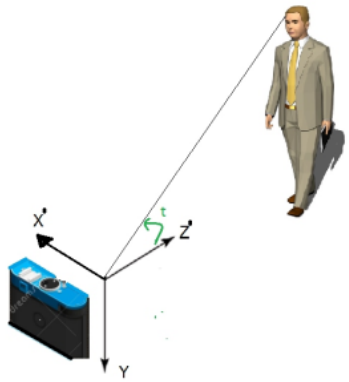
The Openface returns the person's head pose in the camera reference frame as an axis-angle representation as mentioned in the baselines and a position vector localizing the head pose in the image. Also as mentioned, the camera referential is given by Figure 3.1, where the z-axis points to the perpendicular direction to the center of the camera. Imagine a person looking in front of the camera pointing is head to the camera, the Openface returns an angle for the head pose near zero because even if the person is not looking, the head is almost aligned with the z-axis of the camera and only rotating the head these

angles switch.

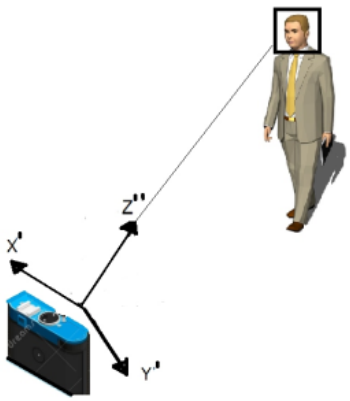
So, having this in mind, to know if the head is in a right pose or not, is essential that the head is aligned with the z-axis of the camera so the Openface can give the head pose angles that have information the head position when a person is in front of the camera. This thinking can be observed in Figure 3.3.



(a) Rotate by a p angle on y-axis.



(b) Rotate by a t angle on x'-axis.



(c) Z"-axis aligned with person's face.

Figure 3.3: Demonstration of how to perform a rotation aligning the head of the person with the z-axis. Adapted from [6].

To accomplish this rotation,i.e to make the camera z-axis point to a person's face is needed the location of the individual (x_p, y_p, z_p). This information can be extracted with the Openface that gives in 3D coordinates in the camera reference frame as said in [6]. Although the camera used in this thesis is an RGB, the Openface can provide 3D coordinates for the head position using algorithms and distances present in the person's face that are similar to everybody. The distance of the head of the person is an assumption by Openface but at distances not very long, like 1 meter, it was found that the values are very reasonable and possible to use. Therefore, the interaction is supposed to be made at 1 meter apart from the robot, giving the value of $z_p = 1m$, however since Openface gives a good value for z in this case, z_p given by Openface is used and with this value is possible to get the rotation that we want to get. Next, are presented the steps to perform the rotation:

1. Knowing this information about the person, and having the Figure 3.3 as a reference, the first step is to calculate the rotation angle p.

$$p = \arctan\left(\frac{x_p}{z_p}\right) \tag{3.9}$$

2. Second, having the p angle, the rotation matrix associated to this angle, this is, associated with the y-axis is calculated.

$$R_p = \begin{bmatrix} \cos p & 0 & \sin p \\ 0 & 1 & 0 \\ -\sin p & 0 & \cos p \end{bmatrix} \tag{3.10}$$

3. As demonstrated in the image above, the next step is to calculate the rotation angle t in the x'-axis to have the z' pointing to the person's head.

$$t = \arctan\left(\frac{y_p}{z'}\right) \tag{3.11}$$

The new z'-axis can be obtained through the next equation:

$$z' = \arctan\left(\frac{z_p}{\cos p}\right) \tag{3.12}$$

4. As done before in the second step, the rotation matrix associated with angle t, this is, associated with the x'-axis is calculated.

$$R_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos t & -\sin t \\ 0 & \sin t & \cos t \end{bmatrix} \tag{3.13}$$

5. Having those rotation matrices, we also need the rotation matrix associated with the head of the person. As said before, the Openface also gives the rotation vector associated with the head, the Euler angles. With these angles is possible to obtain a matrix called R_h with the help of equation 3.6.

6. Having all the matrices needed, the final rotation matrix is given by $R_{hf} = R_h R_t R_p$.

7. This matrix is transformed to euler angles using 3.6. This final vector is given by $h_r = (h_{rx}, h_{ry}, h_{rz})$

The final vector gives a correct and useful information about the head pose.

Head pose classifier

After the detection of the head pose, this is, after the Openface gives the information and all the processes relative to the rotation matrix, R_{hf}, the classification is done. For the classification, the features

are the Euler angles calculated at the end of all the processes described above. The final rotation vector is used in a SVM and a KNN classifier to evaluate the best performance. To train each supervised classifier, datasets were built, where the data had two labels, ”head position to interaction” and ”head position to not interact”. These datasets were videos of people wanting to interact or not wanting to interact, being the Euler angles of the headbox calculated through the method described and saved in the dataset.

3.3.2 Eye gaze module

Salvado [6] used Openface also to obtain the eye gaze. As said before the eye gaze combined with the head position allows to obtain eye gaze at greater distances. The eye gaze is one of the most important modalities in HRI. With that in mind let's move on to the methods used to detect eye gaze.

Eye gaze detection

As explained in the head pose module, the solution used in this thesis is a combination of the eye gaze with the head position, being possible to detect the eye gaze of a person, using Openface, even at long distances. This solution is based on the work described in [6].

The Openface returns two gaze vectors one per eye. They represent where the person is looking at in the camera reference frame and are used as predictors to the direction of the person's look. However these vectors aren't useful for the classification that we want to realize, so we need to perform some calculations to obtain different information using the one given by the Openface, like in the case of the head position. Next are presented, in order, the steps made to get viable information for this module.

1. Having the gaze vectors of each eye, they are normalized to unitary norm.

2. After the normalization we have two vectors for the eye right and left eye respectively, (x_{gr}, y_{gr}, z_{gr}) and (x_{gl}, y_{gl}, z_{gl}), and the angles performed in the x-axis and y-axis by each vector,i.e performed by each eye, are calculated using the next two equations for each eye:

$$a_x = \arccos(x_g) \tag{3.14}$$

$$a_y = \arccos(y_g) \tag{3.15}$$

   The two vectors obtained are (a_{xr}, a_{yr}) and (a_{xl}, a_{yl}).

3. Having the gaze vector angles for each eye is calculated the g vector that represents the mean of the two angle vectors calculated in the step 2. This mean vector represents the gaze angle vector for the gaze of that person since it considers the information of both eyes in a single vector.

$$g_x = \frac{a_{xl} + a_{xr}}{2} \tag{3.16}$$

$$g_y = \frac{a_{yl} + a_{yr}}{2} \tag{3.17}$$

4. As mentioned, the eye gaze is combined with the head pose. The rotation matrix to put the camera directing to the person's head is given by $R_{ef} = R_t R_p$, being these matrices given by the head pose module. With this rotation matrix, the head position vector that localizes the head of the person is rotated with the R_{ef} rotation matrix. The vector from this operation is given by $h_p = (h_{px}, h_{py}, h_{pz})$.

5. Having this final vector as a position vector, we calculate the angles of the vector, for the rotation made by the head after the camera referential are pointing to the person's head. In 2D the angles that are interested in this case are only in x-axis and y-axis. Therefore, first we normalize the vector and after, these angles are given by the next equations:

$$h_{ex} = \arccos(-h_{px}) \tag{3.18}$$

$$h_{ey} = \arccos(-h_{py}) \tag{3.19}$$

The negative sign in the equations is because it is desirable to have a vector pointing from the person's face to the camera and not the opposite. The final vector is given by $h_e = (h_{ex}, h_{ey}, h_{ez})$ and with this vector the difference between g and h_e, dg, are calculated. Before those operations, h_e is normalized.

$$dg_x = h_{ex} - g_x \tag{3.20}$$

$$dg_y = h_{ey} - g_y \tag{3.21}$$

The vector dg gives information about the eye gaze. At long distances, the g vector is given by the person's face direction because of bad detection by the Openface. However, at a small distance g vector is influenced by the direction of the eyes.

6. Having the dg vector which is an angles vector and supposing that the person is at approximately 1 meter from the robot, it is possible to calculate the distance in the image from the center of the camera to the point where the eye gaze is pointed. Like in the head pose module, the interaction is supposed to be made at 1 meter, however, is used the z_P value for the head position as the value for the distance of the person. The next equations give this idea:

$$d_x = \tan\left(\frac{dg_x}{z_p}\right) \tag{3.22}$$

$$d_y = \tan\left(\frac{dg_x}{z_p}\right) \tag{3.23}$$

$$d = \sqrt{d_x^2 + d_y^2} \tag{3.24}$$

The distance from the center of the camera to the point where the eye gaze is pointing can be used in different ways to have a classification about a person looking or not looking to the robot.

Eye gaze classifier

With the distance calculated above is possible to claim that at some distances, small distances,i.e near the camera, the person is looking to the robot. However, at big distances, the person is not looking at the robot. So is the distance a feature we are capable of assembling a classifier based in these outlines. This classifier needs to have a boundary in the distance values that separates the two classes.

A good classifier for this problem, as referred to in chapter 2, is based in a Bayes risk minimization. Having results of persons looking to the robot and not looking to the robot is possible to obtain histograms like the one described in Figure 3.4.

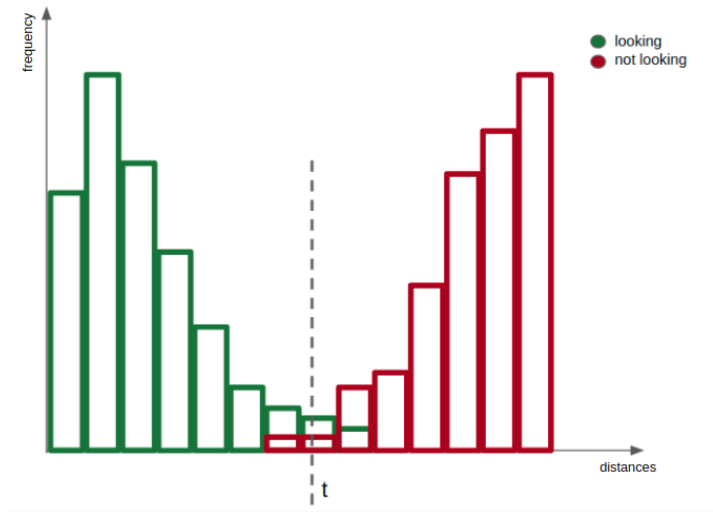


Figure 3.4: Sketch demonstrating a histogram having different values for the distances obtained when a person is looking or not looking to the robot. Also is identified a possible boundary for the classes.

In Figure 3.4 it can be seen that for some distances there may be cases in which the two classes are the output of the classifier. Although the Figure is only a sketch and allows to exemplify a concrete case, is normal for such situations to occur. This is due to detection failures and false positives. For example, in this case, false positives are the cases when at a certain distance the person is not looking at the robot but the classifier considers it to be. The detection failures are the opposite of the previously mentioned. If these two properties exist, the histograms that can be performed, always overlap the data in each class for some values of features (in this case distances). Therefore, in these cases, it is not so easy to calculate a value that allows the division between the two classes, identified by the letter t and outlined in the Figure 3.4.

To calculate this value that allows the division between classes and therefore serves as a classifier, Bayes risk minimization is used. This method seeks to minimize the following equations that describe this particular problem and make it possible to calculate a value for t that optimizes the classification:

$$\min \qquad p_e = p_{df} + p_{fp} \tag{3.25a}$$

$$\min_t \qquad p_e = \int_0^t p_{df}(x)dx + \int_t^\infty p_{fp}(x)dx \tag{3.25b}$$

In the equations (3.25), p_e is the probability error, p_{df} is the probability of detection failures and p_{fp} is the probability of false positives. So these equations try to minimize the probability error, finding the value for t that minimizes the sum of the detection fails probability and the false positives probability, which minimizes the error probability and the classifier is more accurate.

Concluding, based in the Bayes risk minimization, doing a experience where is asked to people look or not look to the robot and saving the distances obtained by the eye gaze module, is possible with the assistance of this method to find a value that separates the two classes, this is, constructing a classifier for the eye gaze.

3.3.3 Posture module

As stated in the Background of this thesis, posture is a key aspect in interaction and is important in its use. A person could be looking to the robot but if the posture is bad, like the person being sideways to the robot in front of it, maybe the person doesn't want to interact to the robot. In a social interaction is not normal or educated for a person trying to interact with someone else in a bad posture.

The posture is also an important aspect for the gestures, because most of the gestures that are used for interaction need a good posture, being possible to not be done correctly if a good posture is not performed by the human.

The posture module performs the detection of the posture of the person and classifies if is posture for interaction or not.

Posture detection

Chapter 2 describes various approaches for the robot to acquire the posture of a human. Considering some aspects, the decision was made to use the Openpose[9][47][48][49] tool to acquire the joints of the human body, that is, the skeleton that can be observed in Figure 2.7.

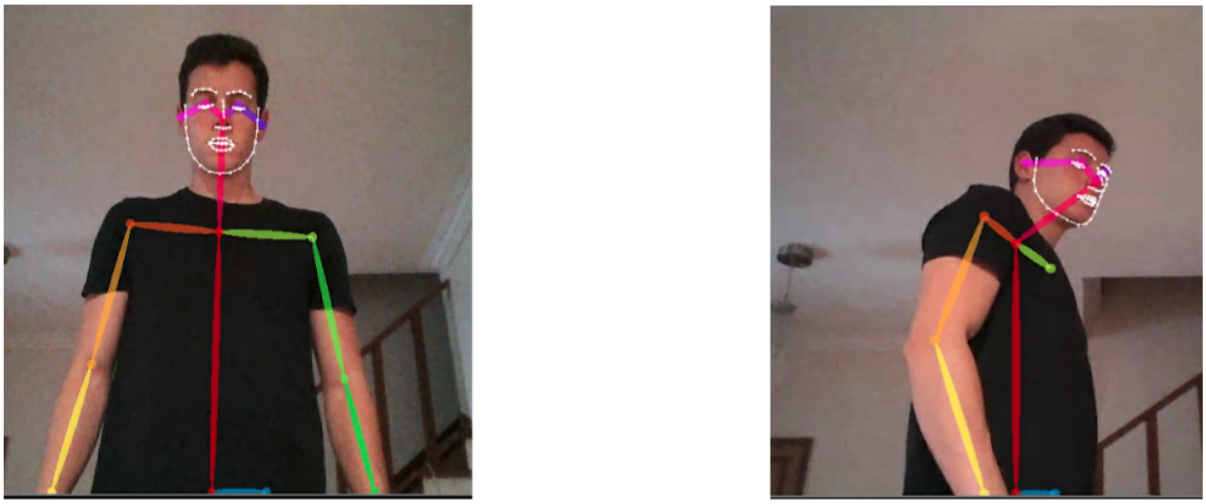
Using this tool it is possible to obtain the location of the joints of the human body in the image, being possible with this information to perform operations on these values. The returned skeleton is in Body_25 format, one of the formats available by the tool and which returns 25 joints of the human body in the skeleton that is calculated for the person identified in the image. The values returned for each joint are the x and y position, in pixels and the image, of each of these 25 joints. Since in this thesis only an RGB

camera is used to obtain the images, the tool only returns 2D values, however, it could also return 3D values for each joint if an RGB-D camera was used.

Supposing the interaction between the human and the robot is near 1 meter apart from each other, like supposed in the eye gaze and head pose module, a feature was considered to infer if the person as a correct or wrong posture to interact with the robot.

The analysis of a person's posture in relation to a posture to interact or not, in this paper, is very much based on how the person's body is rotated against the robot. If a person is facing the robot, even if the posture is not very upright and straight, it is intended that it may indicate a posture to interact with, but a person who is sideways to the robot no longer has a good posture to interact with this one.

Since body rotation is a very important aspect of posture in an intention of interaction, it is possible through the person's shoulders to ascertain posture. Therefore, Openpose is able to return the location of a person's shoulders as they are joints of the human body, and it is possible in an image to identify that a person rotating the body causes the shoulder distance on the x-axis to decrease when the person is not faced to the camera, or increases when the person seeks to be in front and with the chest facing the camera. Those cases can be verified in Figure 3.5. In cases where the person is on his back, this increase also happens however the modalities of eye gaze and head pose serve to exclude this scenario, as they indicate that the person is on his back and that there is no intention of interaction.



(a) Posture to interact.

(b) No interaction posture.

Figure 3.5: Reflect of the human posture in the distance of shoulders.

By calculating the distance on the x-axis between the shoulders, it is possible to obtain a feature that indicates the person's posture towards the robot. (3.26) presents this calculation.

$$dx_{shoulders} = x_{rightshoulder} - x_{leftshoulder} \tag{3.26}$$

This method is possible to implement knowing that the interaction is done at approximately 1 meter from the robot, because the distance of the shoulders identified in the image may vary if a person is close or far from the camera and using only an RGB camera we cannot obtain the depth. So this feature is used existing that assumption.

Posture classifier

After the detection of the posture, this is, after the Openpose gives the information about the joints and the distance of the shoulders being calculated, the classification is done. For the classification, the feature is the distance of the shoulders in the x-axis. This distance is used in an SVM and a KNN classifier to evaluate the best performance. To train each supervised classifier, datasets were built, where the data had two labels, ”posture to interaction” and ”posture not interact”. These datasets were videos of people trying to interact or not according to the posture, being the distance of the shoulders in the x-axis calculated through the method described and saved in the dataset.

3.3.4 Final potential intent module

With the classifications of the 3 modalities (eye gaze, head pose, and posture), a final classification needs to be done. With the values of those classifications, a final potential intent detector can infer if there's maybe intention or not by the human.

To perform this final classification, we need to combine the 3 modalities in one. For this, we can use an approach called score level fusion [50]. This approach fuses the classifications of the modalities like demonstrated in the next equation, giving also different costs for each modality.

$$p_{int} = \alpha(clf_{eyegaze}) + \beta(clf_{pose}) + \gamma(clf_{headpose}) \tag{3.27}$$

In (3.27) the α represents the cost for the eye gaze classification, β represents the cost for the posture classification and γ represents the cost for the head pose classification. The clf represents the value returned by the classifiers of each modality. These classifiers return the score value, that is, a value between 0.5 and 1 if the intent is achieved for that modality or a value between 0 and 0.5 if not. The costs have a value between 0 and 1 and $\alpha + \beta + \gamma = 1$. The final value obtained for this equation is a probability of potential intent.

Knowing this information we can consider the costs as a percentage of importance. So inquiring an X number of people about which modality they think is the most important, we can obtain a percentage value for each modality,i.e the number of people indicated each modality. This values can be passed to probability values and these values are the costs for the intent of interaction detector.

Having all the values to obtain the final probability, a method to classify that probability as a possible intention or a no possible intention by the human can be constructed using a Bayes risk approach like in the eye gaze module. This time we can ask people to interact with the robot and to not interact with him, obtaining values for the eye gaze, pose and head position, and calculating the final probability to the intent detector. After this, saving this results labeled as ”possible intention” or ”no possible intention”, enables to use a Bayes risk minimization approach to obtain a value for p_{intent} that separates the two classes and optimizes this classifier.

3.4 Gestures detector

Once the possible agent's intent has been detected, if it is detected that the individual is a potential individual to interact with the robot, then the robot must proceed to the analysis of the interaction that the human may make. Therefore, the human by wanting to interact and knowing that communication in this work is done non-verbally, the robot expects a gesture to be made by the human to begin an interaction. The interaction between the individual and the robot is done through gestures, demonstrating the importance of building a gesture detector that will be explained in this section, as the gestures that will be implemented in this methodology.

The gestures module aims to identify when a person initiates a gesture. At identifying the beginning of a gesture, this module saves information from the whole gesture until it can infer that the gesture is over, that after a continuous movement there is a stopping time in which the human waits for the robot's response.

The robot having the gesture information performs certain operations to infer what gesture was performed by the human. For this reasoning, the gesture module is divided into processing of the joints trajectory along the gesture, gesture segmentation, classification of each segmentation and final classification of the segment sequence of the movements that originate the gesture.

The methodologies used for this module and each subsection are outlined and explained below.

3.4.1 Types of gestures

In the context of this master thesis, the types of gestures that are wanted to be detected are interactional gestures as described in section 2.1.3. In this work, the gestures to analyze are only dynamic gestures, as an improvement in the work of [6].

In social interaction, there are important gestures to start an interaction, like a handshake or waving the hand. These are gestures that want a return message if they were first made. These types of gestures are those that interest us to study because in this work the aim is that the robot after inferring if there is an intention of interaction, having an intention, he analyzes the gesture that the person performs, that is, the person interacts with the robot, wanting a robot interaction in return. It is this type of interaction that is intended to be studied and which is used as a basis for choosing some gestures used for the gesture detector, as the robot needs a-priori to know which gestures can be made by the human.

The gestures chosen are handshake with any arm, waving with an arm, a bow and a praying position with hands.

3.4.2 Dynamic gestures analyzer

The dynamic gesture analyzer combines all the methodologies used to detect and identify the gestures performed by the individual. To facilitate the explanation of the methodologies, this subsection is divided into two. The first one refers to the methodologies used to obtain gesture information (this information comes from the joints of the human body), filter the trajectory of the joints and segment the gesture. The

second refers to the methods used in the classification of each segment and consecutive classification of the sequence of segments, the gesture, and at the end of all processes, it is expected that the robot can infer which gesture was performed to interact correctly with the individual.

Gestures segmentation

To obtain information on the gestures that the human is performing, Openpose is used, as used in the posture module detection. Openpose allows you to get the 2D location in the RGB image of 25 joints along the human body. With this information and having a sequence of images that represent the gesture being performed continuously, it is possible to grab the trajectory that the hands, arms, and trunk make along with the frames. These are the 3 areas of the body that interest us to analyze, taking into account the gestures that are made known a-priori to the robot, and that can be performed by the individual who wants to interact.

As mentioned earlier in Chapter 2, a gesture is often made up of several motions.In this work, this view is adopted to make a segmentation of gestures based on accelerations,i.e when we move from one arm movement to another it is possible to denote that at the end of the first movement we finish this and perform the second right afterward but differently because it is a distinct movement. This idea can be seen in Figure 3.6 where different segments of a wave gesture are demonstrated. So for each movement, there is a local maximum acceleration, since we start from a resting situation to a situation where we perform a movement. When calculating the local maximum of each gesture movement, between these two maximums there is a local minimum that is due to the movement from one movement to another, that is, when a movement is being completed the acceleration decreases and only increases again when the second movement begins to be performed.

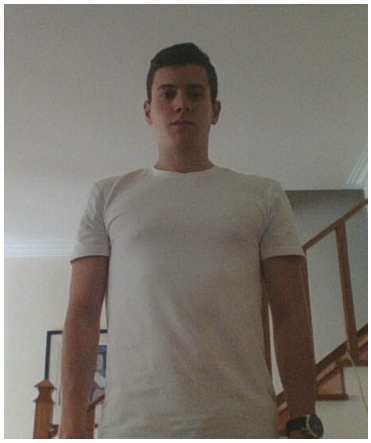
Based on these assumptions, the segmentation of the gesture is made. Through the acceleration of the joints and their maximum and minimum local accelerations throughout the gesture, it is possible to do the segmentation of the gesture. Next are presented, in order, the steps for the segmentation.

1. Acquisition

   Obtain the 2D body joints localization using Openpose. Knowing that the Frames per Second (FPS) of the Openpose are equal to 10, this means that from a frame to another 0.1 seconds are passed. So with this information and the positions of the joints in the image, is possible to calculate for each joint the acceleration from one frame to another.

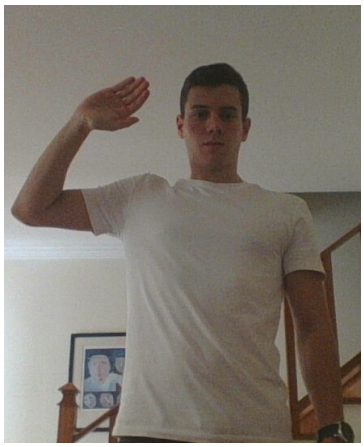
2. Gesture segmentation

   From one frame to the next, if the hands, arms, and trunk joints positions have big differences, start saving the information of joints for each frame in a data structure. This information is used to calculate the joints accelerations. Accelerations that come from small movements or lack of precision through the frames when the person is still are not considered. There's a filter that does this analysis, with the difference of pixels between two same joints, at two sequential points in time, it is possible to set a minimum value to start calculating the accelerations. This value in this work

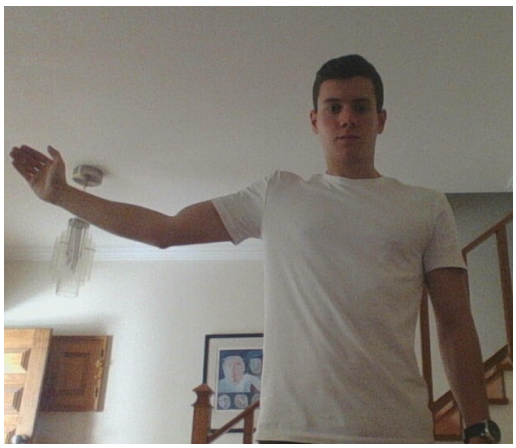


(a) Rest position.

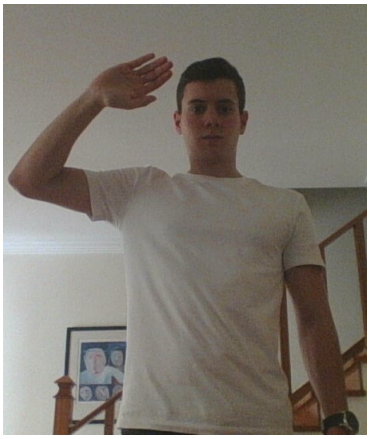
(b) Raising the arm.



(c) Moving the arm from the wrist to the elbow for the right.



(d) Moving the arm from the wrist to the elbow for the left.



(e) Moving the arm from the wrist to the elbow for the right.

Figure 3.6: Possible segmentation of a wave gesture.

is 30 pixels for all joints less than the neck joint which is 20 pixels. The information stops to be saved when the accelerations of all joints are zero for 2 seconds or more.

3. Computation of accelerations

With that information a matrix of joints accelerations can be constructed. In this matrix, the columns are the joints (j is the total of joints used) and the lines are the frames (n is the total of frames). In equation 3.28 this matrix is represented. In (3.29) is also represented for a specific joint (m) the accelerations vector for that gesture that is being analyzed.

$$\begin{bmatrix} \frac{\mathrm{d}^2 x_{11}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{12}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{13}}{\mathrm{d}t^2} & \dots & \frac{\mathrm{d}^2 x_{1j}}{\mathrm{d}t^2} \\ \frac{\mathrm{d}^2 x_{21}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{22}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{23}}{\mathrm{d}t^2} & \dots & \frac{\mathrm{d}^2 x_{2j}}{\mathrm{d}t^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{d}^2 x_{n1}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{n2}}{\mathrm{d}t^2} & \frac{\mathrm{d}^2 x_{n3}}{\mathrm{d}t^2} & \dots & \frac{\mathrm{d}^2 x_{nj}}{\mathrm{d}t^2} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nj} \end{bmatrix} \tag{3.28}$$

$$\begin{bmatrix} a_{1m} & a_{2m} & a_{3m} & a_{4m} & \dots & a_{nm} \end{bmatrix} \tag{3.29}$$

Having for each joint an acceleration vector, it is possible to segment the gestures through the calculation of the local maximums and minimums, however, this vector may not be very smooth having large discrepancies. So a method to filter the set of values of this vector could be implemented.

4. Smoothing

   The filter implemented by using a convolution of a scaled window with the signal. So, for a signal as this one, the signal is prepared by introducing reflected window-length copies of the signal at both ends.Different windows can be used to smooth the signal and in Figure 3.7 some possible smooth windows can be seen. This Figure was adapted. [1]

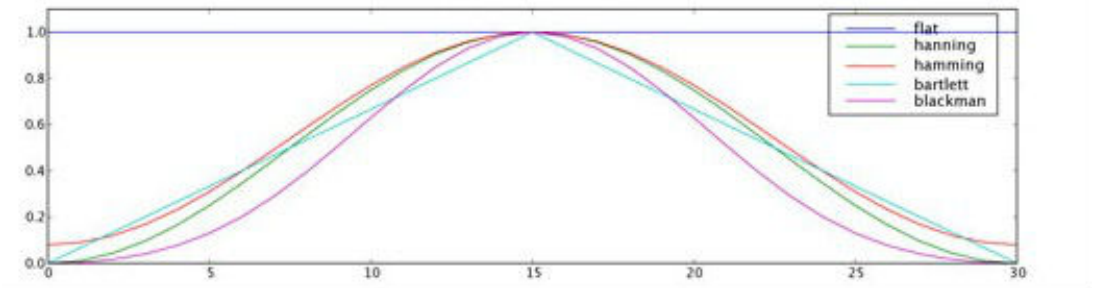


Figure 3.7: Some smoothing windows for the convolution of the accelerations vector.

5. Finding local maximums and minimums

   Find the local maximums and minimums and get the frames of these points for each joint.

   Have for each joint a vector containing, in ascending order of frames, the frames of local maximums and minimums. In (3.30) this is represented, where f_{begin} represents the beginning gesture frame, f_{end} represents the end gesture frame, f_{min} represents a frame where a local minimum was found and f_{max} represents a local maximum frame.

$$[f_{begin} \quad f_{max} \quad f_{min} \quad f_{max} \quad f_{min} \quad f_{max} \quad \dots \quad f_{end}] \tag{3.30}$$

6. Movement segmentation

   The number of local maximums indicates the number of segments that were capable to make to the gesture. The sequence of the final vector presented above is always like that, with a minimum after a maximum and a maximum after a minimum.

With this information, we are capable to segment the gesture and use those segments in different ways to obtain a classification to each one of them.

Gesture segments classifier

Having the vector described above, which gives the frames for the segments of the gesture, using this information and the 2D points of the joints along with the gestures saved, the gestures segments

[1] The description of this method can be seen in https://scipy-cookbook.readthedocs.io/items/SignalSmooth.html. This website was accessed at 10/10/2019.

classifier can be built. With the information acquired through gesture segmentation, it is possible to use it in several ways, being able to classify it and give classes for segmentation. Therefore, with a segmented gesture, the robot wants to know what kind of segments it has detected. These segments are, for example, those presented by the Figure 3.6 and others that allow generating the other gestures that are used in this thesis, and that have been previously explained. In the case of a handshake, the gesture consists of only one segment, raising the arm towards the robot, however, a bow is made up of two segments, the human first bending the trunk towards the robot and then returning to the starting position.

By giving classes to these segments, a machine learning algorithm can be used, that allows the robot to classify these segments and ultimately decide, in their order, what kind of gesture the individual is performing. It is based on this thinking that certain methodologies have been taken for the gesture classifier.

As input information for the classifier are used the angles that certain joints make with others from the beginning of a segment to the end of it. Imagine, in the case of handshaking, when we perform the only segment of this gesture which is raising the arm towards the robot, the joint at the wrist with the joint at the shoulder of the arm that performs the movement generates an angle. This angle can be used as an input feature in the classifier. Therefore, as input information in the segment classifier, angles between joints are used. However, based on the gestures to be studied, it can be seen that 5 angles are important and used to identify any segment of these gestures. In Figure 3.8 these angles are presented.

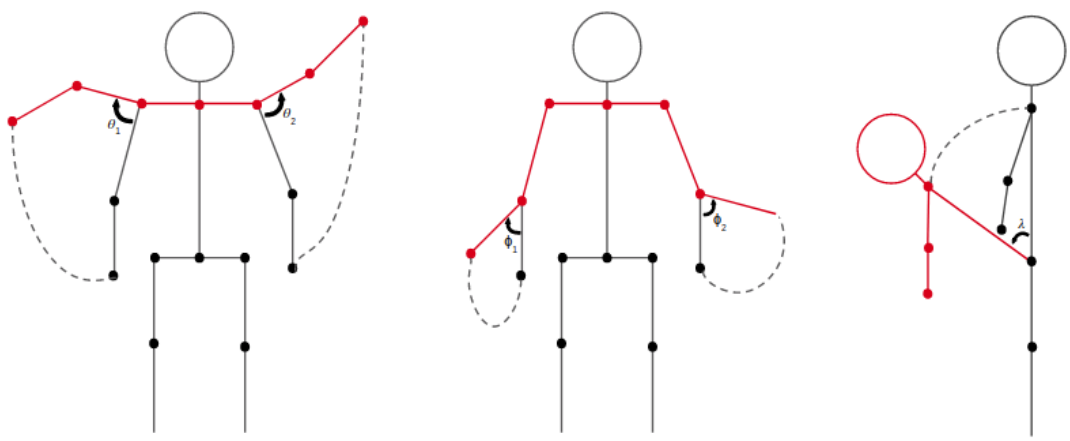


Figure 3.8: The 5 angles used in the segments of the handshake or wave done by one of the arms, the bow and the praying position with the hands.

These angles are sufficient to classify the segments of the gestures that are studied in this thesis. If we do not consider the angles that are repeated for the other arm, θ_2 and ϕ_2, 3 angles define the gestures. The angle θ_1 relates wrist joint to the shoulder joint, thus indicating whether the arm is raised or lowered and the position of the wrist relative to the shoulder. It also reflects the wrist motion to the right or left of the shoulder. This angle is calculated by:

1. Calculate the distances for the x-axis and y-axis from the joint wrist to the shoulder joint and use

these distances as the 2D point of the wrist. This is calculated for the points of the begin and end of the segment. With this, the shoulder is the referential.

2. Normalize the new points for the wrist joint at the begin and end of the segment.

3. Calculate the slopes, m_1 and m_2, of the wrist joint and the angle made by it, $wrist_{angle_1}$ and $wrist_{angle_2}$, for the begin and end of the segment, with the next equations:

$$m_1 = (x_{wrist\_begin}/y_{wrist\_begin})) \tag{3.31}$$

$$m_2 = (x_{wrist\_end}/y_{wrist\_end})) \tag{3.32}$$

$$wrist_{angle_1} = \arctan(m_1) \tag{3.33}$$

$$wrist_{angle_2} = \arctan(m_2) \tag{3.34}$$

4. With these two angles, the difference gives the final angle,i.e, the angle made by the wrist relative to the shoulder in one segment.

$$\theta_1 = wrist_{angle_2} - wrist_{angle_1} \tag{3.35}$$

The same is done for θ_2 with the difference that the joints for the wrist and shoulder are the ones from the other arm.

ϕ_1 is the angle that the wrist makes with the elbow, this let us know if the arm moves between the wrist and the elbow, for example when waving, where the wrist moves but the elbow remains still. Therefore, this angle seeks to obtain information on horizontal wrist movements relative to the elbow. The next items describe the solution.

1. Calculate the distances for the x-axis and y-axis from the joint wrist to the elbow joint and use these distances as the 2D point of the wrist. This is calculated for the points of the begin and end of the segment. With this, the elbow is the referential.

2. Normalize the new points for the wrist joint at the begin and end of the segment.

3. Calculate the angle made by the wrist joint, $wrist_{angle_3}$ and $wrist_{angle_4}$, for the begin and end of the segment, with the next equations:

$$wrist_{angle_3} = \arctan 2(x_{wrist\_begin}, y_{wrist\_begin}) \tag{3.36}$$

$$wrist_{angle_4} = \arctan 2(x_{wrist\_end}, y_{wrist\_end}) \tag{3.37}$$

The use of $\arctan 2$ is because we want to grab the information from horizontal movements in this case.

4. With these two angles, the difference gives the final angle,i.e, the angle made by the wrist relative to the elbow in one segment.

$$\phi_1 = wrist_{angle_4} - wrist_{angle_3} \tag{3.38}$$

The same is done for ϕ_2 with the difference that the joints for the wrist and elbow are the ones from the other arm.

Finally, lambda gives information about the angle that the neck makes with the waist (end of the trunk), indicating whether a person is lowering the trunk forward or rising to stand straight. Like in the other angles the steps are described.

1. Calculate the distances for the x-axis and y-axis from the joint neck to the waist joint and use these distances as the 2D point of the neck. This is calculated for the points of the begin and end of the segment. With this, the waist is the referential.

2. Normalize the new points for the neck joint at the begin and end of the segment.

3. This gesture is divided into two phases, the lowering of the trunk and the ascent. With a 3D camera, it would be possible to easily calculate the angle to be calculated along any of these movements. However, only one 2D camera is available.

   So, we can suppose that this gesture when performed, the first frame in a downward movement represents the person standing, just as in the last frame in a rising movement, the latter represents the person standing. With this information, we can always calculate the distance from the end of the trunk to the person's neck using the y-axis. Since bowing motion is expected to be associated only with raising and lowering the trunk towards the robot, only the information on the y-axis is used. With this, we can use trigonometry to get the λ angle. It is possible to imagine in the Figure 3.8, for the case of the bow and for the angle λ, the triangles that are created with the hip as the reference center, for the movements performed with the trunk. Next, are presented the equations to solve this problem.

   (3.39) and (3.40) represent the downward movements. If the movement does not exceed the hip, λ is given by (3.39). However if it passes, λ is given by (3.40).

$$\lambda = \arccos\left(\frac{y_{neck\_end}}{y_{neck\_begin}})\right) \tag{3.39}$$

$$\lambda = \pi - \arccos\left(\frac{y_{neck\_end}}{-y_{neck\_begin}})\right) \tag{3.40}$$

   The negative sign in $-y_{neck\_begin}$ is due to the fact that this solution uses distances, and in the case that the neck exceeds the hip in the movement made, the value of y_{neck_end} is positive and

not negative, because the reference used is the center point of the hip. For (3.39) the two values are negative and in a fraction, the negative signs are canceled so there's no need to do the same as in this case.

For the rising movements, the idea is the same. (3.41) and (3.42) represent the rising movements. If the movement does not start under the hip, λ is given by (3.41). However if it is underneath, λ is given by (3.42).

$$\lambda = \arccos\left(\frac{y_{neck\_begin}}{y_{neck\_end}})\right) \tag{3.41}$$

$$\lambda = \pi - \arccos\left(\frac{y_{neck\_begin}}{-y_{neck\_end}})\right) \tag{3.42}$$

The negative sign in $-y_{neck\_end}$ is for the same reasons as mentioned above.

In the end, for each segment, a vector with these 5 angles is returned so it's possible to use as a feature for the classification. This vector is described in (3.43).

$$[\theta_1 \quad \theta_2 \quad \phi_1 \quad \phi_2 \quad \lambda] \tag{3.43}$$

For the classification of the segments, an SVM and a KNN were studied, using as label all the possible segments from the gestures mentioned in this thesis. To train the SVM, people were recorded doing the gestures defined above, and for each gesture, a segmentation was done and calculated the equation (3.43) vector for each segment, being this one labeled according to the expected segment.

Gesture sequence classifier

After the classification of all the segments identified in a gesture, the gesture is predicted. To predict the gesture, as mentioned in chapter 2, using a Hidden Markov Model for each gesture, it is possible to obtain a probability of a sequence of classified segments being a certain gesture. So the input of the HMM are the segments of the gesture being analyzed, as a sequence, in ascending temporal order, and the output is the probability of that sequence be the gesture described by that model. Having one model for each gesture and having 6 gestures studied in this master thesis, we need 6 HMM, one for each gesture. The data to train these HMMs are possible sequences of classified segments for each gesture. This data were obtained recording people and the sequences of classified segments when ask to do one of the 6 gestures.

In conclusion, the gestures classifier classifies the segments after the gestures segmentation and uses that classifications as a sequence, so a HMM can give the probability of that sequence is one of the 6 gestures studied. With all this, the gestures are detected and classified, so the robot could correctly interact with the individual.

Chapter 4

Implementation

After discussing the methodologies used in this thesis, it is also necessary to discuss the entire implementation of the presented solutions and how to perform all processes referred to a real case. The implementation of a work is everything that is done to obtain the information needed, to transfer this information from one side to another and the real and visible and real answers to what the methodologies intend to accomplish.

This chapter presents the general architecture of the entire implementation of the tool to be proposed in this work, all the sensors used to obtain information, the media, the structure of the built software and the control of all robot behavior for this tool. interact with the human in order to lead the human to identify social interaction with the robot, something close to interaction with another human.

4.1 General implementation

A diagram showing how the processes used are connected and how they are distributed is presented in Figure 4.1.

The architecture was designed as follows: the information is obtained through the sensors present in the robot, this information is then used in the methodologies already presented, for the potential intent detector and the gesture detector, and in the end the robot interacts with the human-based on the gesture performed by this one. The robot does not have a great computational power, which is required for the built solutions, so these computational processes have to be performed on a server. With the existence of a server, communication is an aspect that must exist and is therefore visible in the described architecture. This communication intends to send the data acquired by the robot to the server and, after processing the information, it is intended that the server send the final result regarding the interaction, ie, if the human wants to interact and what kind of gesture the individual performed.

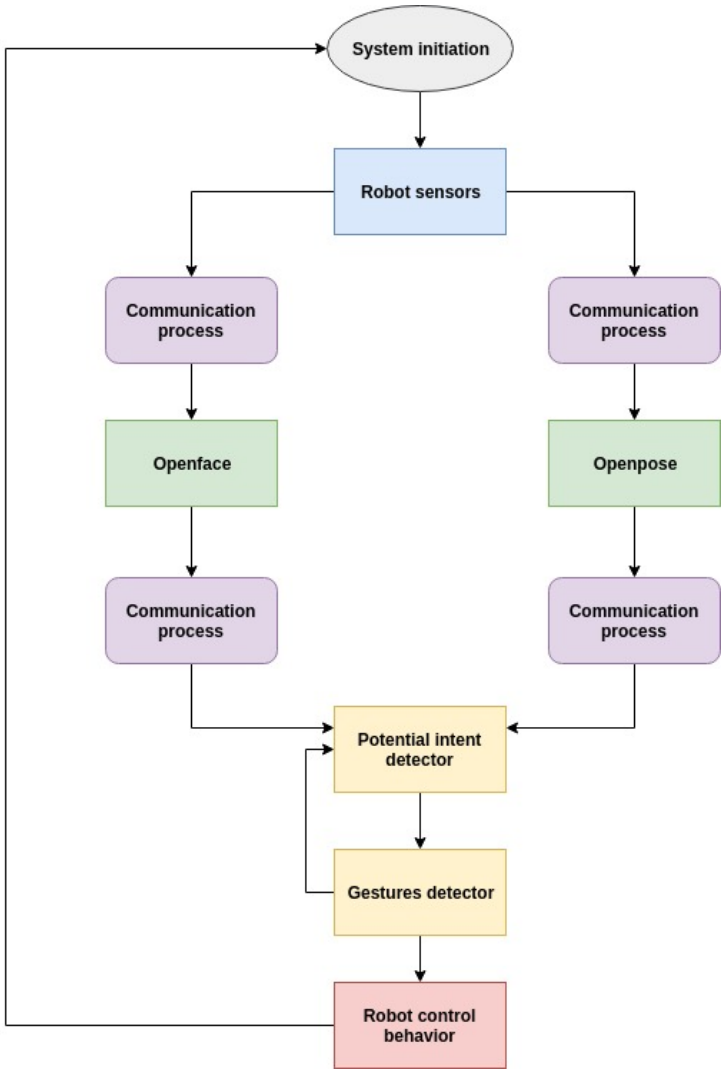


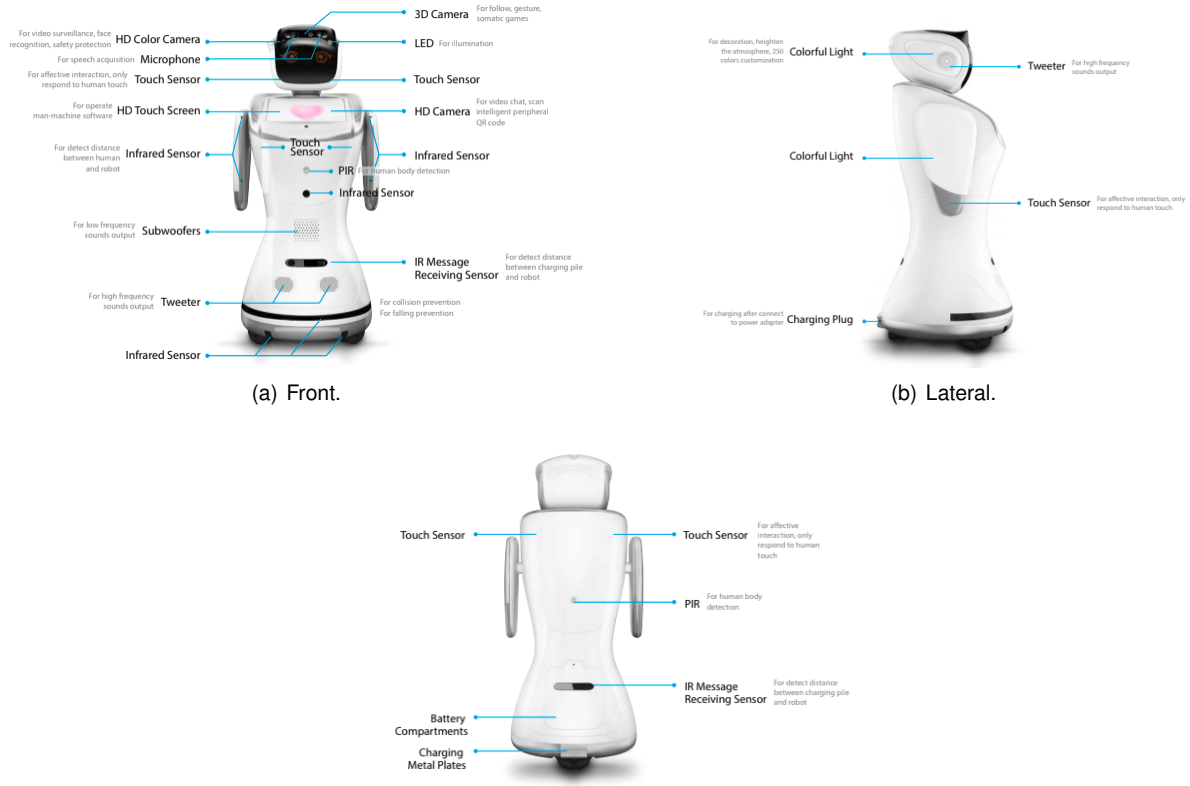
Figure 4.1: Architecture for the general implementation.

4.2 Robot sensors

The robot used in this work is Sanbot. This robot has several sensors that can be used to acquire various types of information from outside the robot. In addition to this aspect, the robot also seeks to have a sociable and friendly aspect. In Figure 4.2 the robot is presented in three different views, front, side and back, and each of these views identifies and explains the sensors and some elements of the robot's physical appearance. These Figures and information were taken from the Sanbot user manual [51].

Although the robot has a 3D camera, in the context of this master's thesis the objective is to obtain the information through an RGB camera. Therefore, the robot sensors that are used in this work are the RGB camera in the top of its head and the touch sensors on the end of the robot arms.

To obtain the individual information these are the sensors used, however, some robot instruments are also used. These instruments are the sound made by the robot (speech), and the tablet that the robot has on its chest for image demonstration. The use of these instruments is intended for the behavior that the robot makes towards the gestures made by the person.



(a) Front.

(b) Lateral.

(c) Back.

Figure 4.2: Sanbot appearance and sensors. Sub-Figures adapted from [51].

4.3 Communication process

As mentioned earlier, using a server is essential for the implementation of all models and processes used and built, as the robot does not have sufficient computational power.

Communication is done through Robot Operating System (ROS) which is a collection of software frameworks for robot development, providing standard operating system services.

The image captured by the robot camera can be sent via a ROS node, and there is another node on the server that receives this information. The server, when receiving a data packet, decompresses it obtaining the image captured by the robot's camera. Having the camera image, it is intended to use the detection tools for the head, face, and posture of the person identified in the RGB image. To use these tools a ROS wrapper is used for Openpose [1] and another for Openface [2]. These ROS wrappers are just ROS-based services that allow the server to subscribe to the RGB image, run the tools (Openpose and Openface) for that image separately, and publish the values that should be returned at the end of these tools. These returned values are published to the network to which the server is connected, and it is then possible through another ROS subscriber, built into the server to receive the information of the tools and to perform all processes associated with the methodologies. The two wrappers work at different frequencies, Openface has 15 FPS and Openpose has 10 FPS. To solve this problem, a

[1] The description of the ROS wrapper for Openpose is in https://github.com/firephinx/openpose_ros. This website last visit was at 12/10/2019

[2] The description of the ROS wrapper for Openface is in https://github.com/ditoec/openface2_ros. This website last visit was at 12/10/2019

mutex was implemented in the server subscriber, so the Openface results are faster, but only when there's Openpose values returned to, the processes are performed. Finally, given that the person wants to interact and that person interacts by making a gesture, the intention and gesture information read by the robot is published on the network, as before, through a ROS publisher. In the robot, there is a ROS subscriber that receives this information and lets the robot know what to do with the intention of interaction and the gesture that the person performed.

The pipeline for all of this process is represented in Figure 4.3.

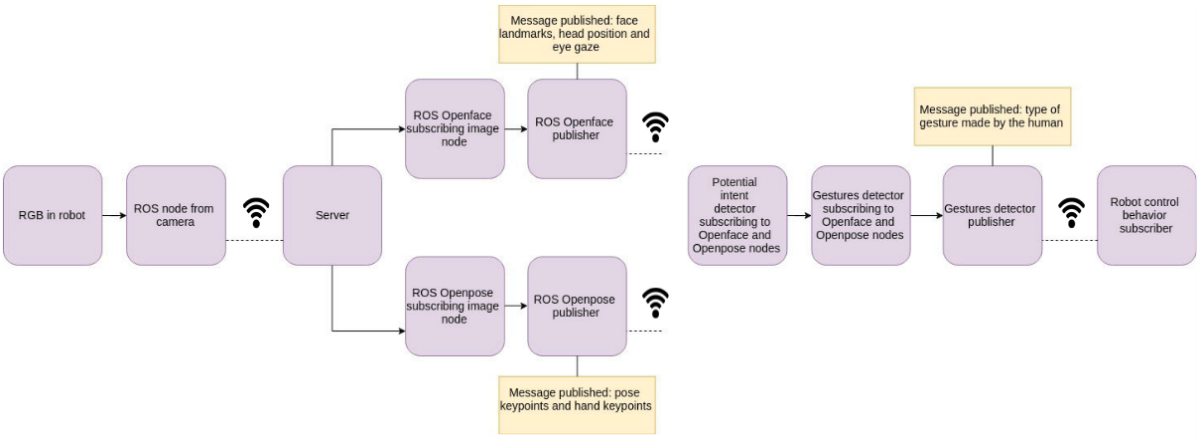


Figure 4.3: Pipeline for the communication process.

4.4 Representation of the processes operation

The intended tool in this work is based on two modules, the potential intent detector, and the gesture detector. That said, it is important to make a representation of these two processes, ie, how they were implemented and how their operation is performed.

4.4.1 Potential intent detector

The potential intent detector infers if there's a potential intention, from the person, to interact with the robot. Knowing the methodologies for this module, in Figure 4.4 is represented the operation of this process.

4.4.2 Gestures detector

Finally, the gestures detector infers which gesture was made by the person, from the gestures known by the robot a-priori. Knowing the methodologies for this module, in Figure 4.5 is represented this process operation.

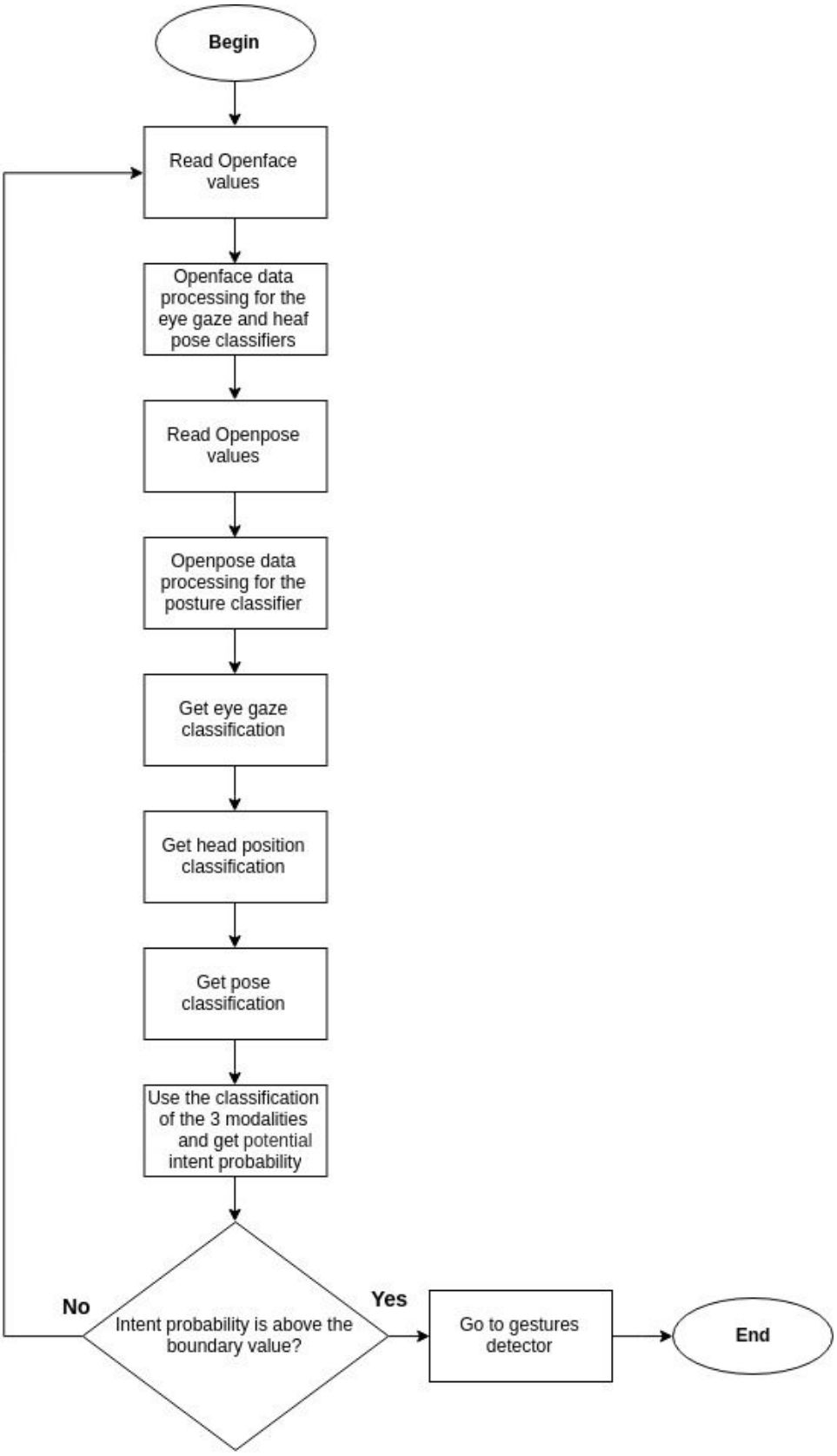


Figure 4.4: Fluxogram for the potential intent detector.

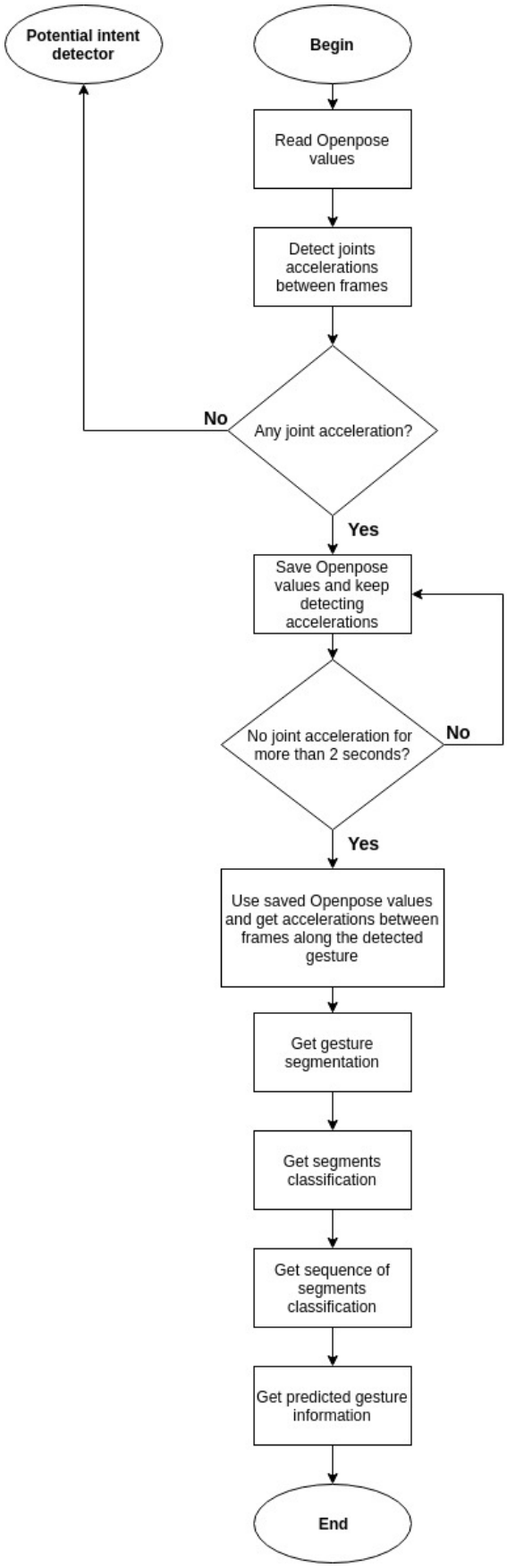


Figure 4.5: Fluxogram for the gestures detector.

4.5 Control behavior of the robot

Control of robot behavior is explained in this subsection. It is important to note that in this case, I'm talking about all the behavior that the robot must do from the moment the created tool is started. Earlier, when this control was mentioned during this chapter, as for example in the Figure 4.1, only part of the robot's behavior was referred to, that is, from the moment the robot receives the gesture information performed by the individual.

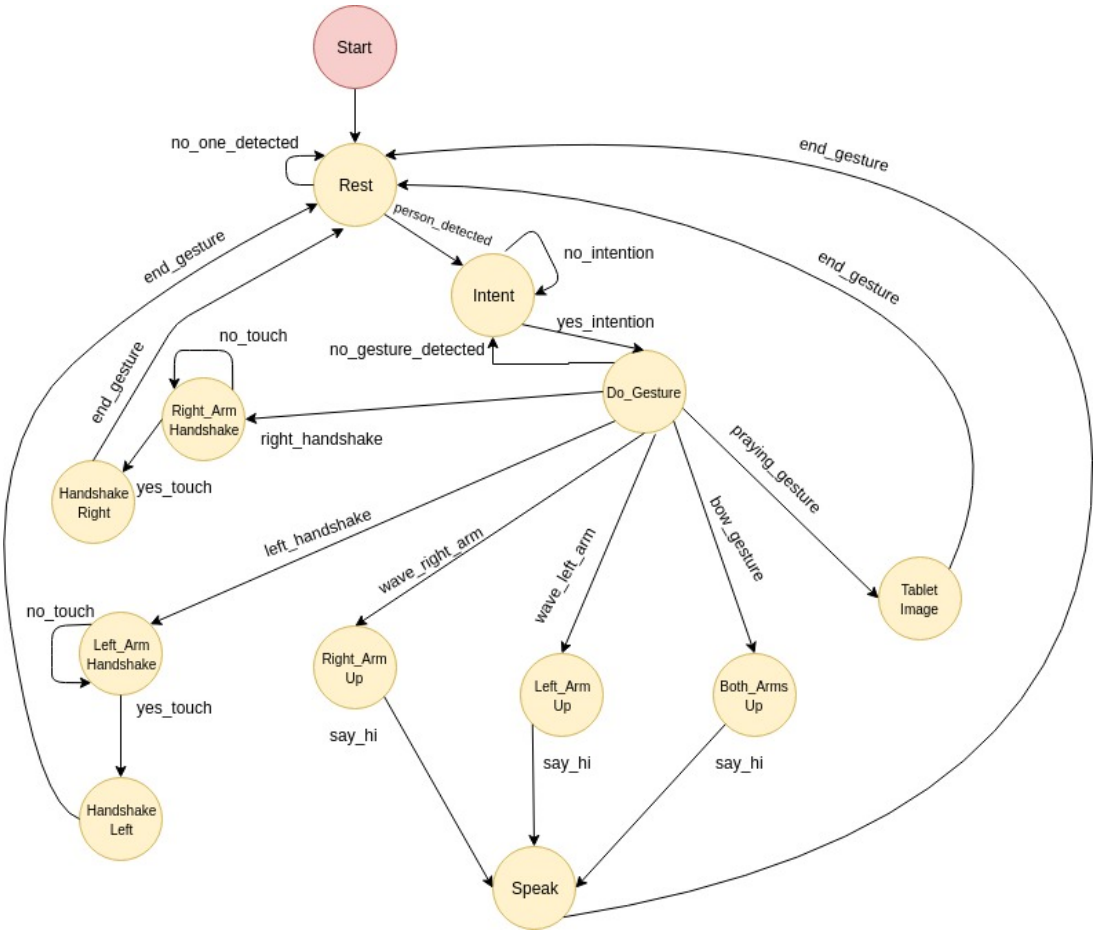


Figure 4.6: State machine of the behavior of the robot.

Figure 4.6 shows the state machine that represents the general behavior of the robot proposed. In this Figure, it is possible to observe that when starting the tool, the robot continuously seeks to detect a person present in the image (**Rest** state). If it detects a person, the robot continuously seeks to know if that person is able to interact with the robot or not (**Intent** state). If the person is able to interact, the robot waits for this person to do a gesture (**Do_Gesture**), and when that person performs one of the 6 gestures the robot knows, each gesture leads to a different state transition.

In the case of handshakes, according to the handshaking arm, the robot raises the arm to a handshake position and waits for it to be touched by the individual (**Left_Arm Handshake** or **Right_Arm Handshake** states). If the individual does not touch the hand raised by the robot, the robot keeps waiting for that, and if the individual touches, the robot waves its arm in a handshake movement (**Handshake Left** or **Handshake Right** states, depending on the arm). In the end, the robot returns to the state **Rest**.

In the case of waves, according to the waving arm, the robot raises the arm to a waving position (**Left_Arm Up** or **Right_Arm Up** states). Then the robot speaks, saying the expression ”Hi” (**Speak** state). In the end, the robot returns to the state **Rest**.

For bowing, the robot raises both arms up (**Both_Arms Up** state), then says ”Hi” (**Speak** state). After this last state the robot returns to the **Rest** state.

Finally, if a praying position is performed, the robot displays an image associated with this symbol on its tablet which is present on the chest (**Tablet Image** state). It then returns to the **Rest** state.

So all the behavior of the robot can be given by the state machine present above.

Chapter 5

Experiments and Results

This chapter presents the experiences needed to gather data to train the classifiers and test them. In addition to the experiments, the results are presented, discussing what would be expected and what can be observed in what was obtained.

5.1 Data collection, training, and testing classifiers

All modules in this work receive information as input and classification information, so each module has a classifier that indicates, for example in the case of eye gaze, whether a person is looking at the robot or not. These classifiers need to be trained and then their accuracy must be tested so that it is possible to choose the best classifier for each module when conducting a study comparing the performance of each classifier.

To perform these two processes, data-sets with specific information for each classifier to be trained are required. These data-sets were all built/recorded, as the information is obtained only by an RGB camera, and different numbers of people were used in the experiments performed.

5.1.1 Processes to train and test the classifiers

There are 4 types of classifiers studied in this thesis: the KNN, the SVM, the Bayes risk minimization and the HMM. The Bayes risk minimization was implemented for the eye gaze and for the potential intent detector, so a boundary value could be found to have a classifier. To find that boundary, data recorded with people looking and not looking, or wanting to interact or not, were used. For the KNN and the SVM, to train the classifiers and evaluate the performance, was used a method called cross-validation. In this method was used 20% of the data-set to test and 80% of the data-set to train, divided in 5 folds, which can be observed in Figure 5.1. For the HMM, the models were trained with built data-sets that indicate the possible sequences of classes for each gesture. The test was done with real data recorded and to evaluate these models a confusion matrix was done.

For the SVM algorithm, a range of C between 0.01 and 100 and a range of γ between 0.01 and 100 were considered. For the KNN algorithm, a k between 1 and 10 was considered

Figure 5.1: Visualization of the effect of a cross-validation in a data-set with 20% to test and 80% to test, divided in 5 folds.

5.1.2 Experience 1 - Inquiry people to choose gestures and obtain costs for final intent module

Objectives

This experiment aimed to find out to some people what kind of gestures they are more likely to use to initiate an interaction and which modality (eye gaze, posture, head position) they find most important for a potential intent detector. Also were asked some questions to know the opinion of different people about problems related to this thesis.

Description

A questionnaire was made in order to obtain from some people some important information related to the subject addressed by this thesis. The two main topics of information that were intended with this questionnaire were the gestures that people find most likely to use to initiate an interaction and which modality they considered most important, in order to infer if a person is able to interact with the robot and is considered a potential intention of interaction for the robot. The modalities presented for choosing were those used in this thesis, eye gaze, head pose, and posture. 32 people were surveyed.

Results and discussion

This was the first experiment performed in this thesis. This experience made it possible to make decisions that were made in the methodologies of this thesis as explained in previous chapters.

As main topics, the ones presented in the experiment description are the most important and the reason for the creation of this questionnaire, however questions regarding the robot's appearance and

the location of the robot that most captivated people's eyes were also asked. From these questions, it was concluded that practically everyone thought that the robot's appearance led people to want to interact with it and to feel comfortable due to its "loving" aspect that some people mentioned. As for the place that most captivated people's eyes, 61.5% said it was the robot's face, 30.8% it was the tablet on the robot's chest, and 7.7% it was the arms.

Regarding the type of modality that people think is most important for a potential intent detector the following results were obtained: 47% think the direction of look is the most important, 38% think it's the posture and 15% think it's the position from the head. It should be noted that the questionnaire explained what a potential intent detector was intended to do. These results allowed me to give values to the costs in (3.27) associated with the potential intent detector: α = 0.47, β = 0.38 and γ = 0.15.

Finally, the gestures indicated by the people who performed the questionnaire were the static and dynamic wave and the handshake. In addition to these gestures, it was also considered the gesture of a bow used in other cultures as a greeting and a praying gesture in order to be able to analyze a gesture that uses the movement of both arms simultaneously.

5.1.3 Experience 2 - Record data for gaze classifier and construct the classifier

Objectives

This experiment aimed to acquire data from people looking and not looking at the robot. With this data, it was intended to train the eye gaze classifier and evaluate its performance.

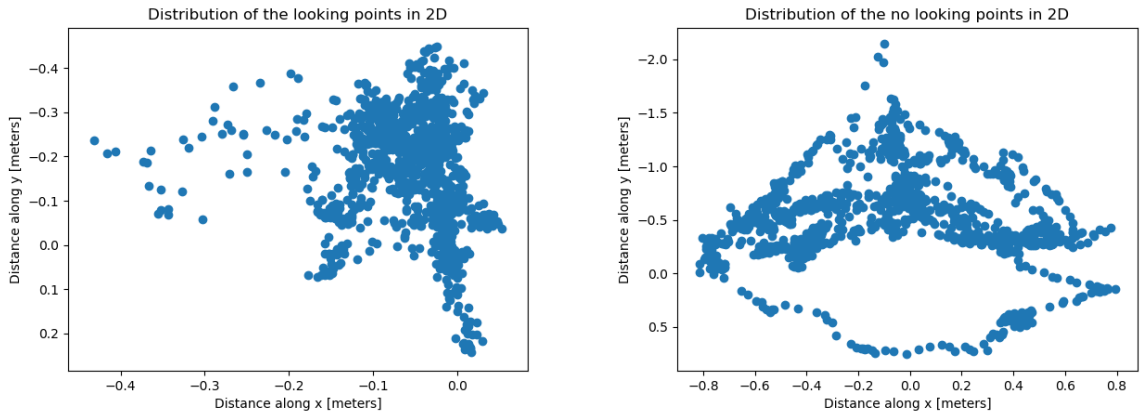
Description

Four people, two male and two female, were asked to stand within 1 meter of the robot. Then they were asked to look at the robot as if they wanted to interact with it. They were also asked to do the opposite, not to look at the robot. Finally, the eye gaze classifier was trained and tested.

Results and discussion

In Figure 5.2 is possible to visualize the results obtained for the situations in which individuals tried to look at the robot or not. It is important to note that the axes in this Figure are in accordance with the camera frame. In the case that was wanted to look at the robot with the intention of interacting with it, it is clear from the results of sub-Figure 5.2(a) that the values are close to zero. The value of zero is when the person looks straight into the robot's camera and it can be said that most people looked closer to zero values, such as at a distance of 0.2 meters. This is because an individual seeks to look at the robot's face or body to interact with it, and these locations appear slightly below the camera's location, as it is located at the top of the robot's head. However, most values are above the camera's location, which is because the robot is low and most people have to look down. As noted above, the eye gaze module combines the head position at long distances, and the robot being shorter, most people do not fully lower their heads which gives these values around the position of the camera a little above it.

In the event that people were asked not to look at the robot, the results in the sub-Figure 5.2(b) show that the distances to the camera are high and near 1 meter, and most choose to look up or sideways, because of the robot being shorter than most people. It should also be noted that a person looked at the floor and it is possible to observe this case in the results.



(a) Distribution of the 2D points of the eye gaze when a person is looking to the robot.

(b) Distribution of the 2D points of the eye gaze when a person is not looking to the robot.

Figure 5.2: Location of looking points, (a), and not looking points, (b), for the eye gaze.

With these values obtained in this experience, was possible to construct a histogram with the distances of the looking points and the no looking. Having this histogram represented in Figure 5.3, the Bayes risk minimization was implemented to find the value that gives the eye classifier the ability to get a prediction of the eye gaze having a distance as an input value.

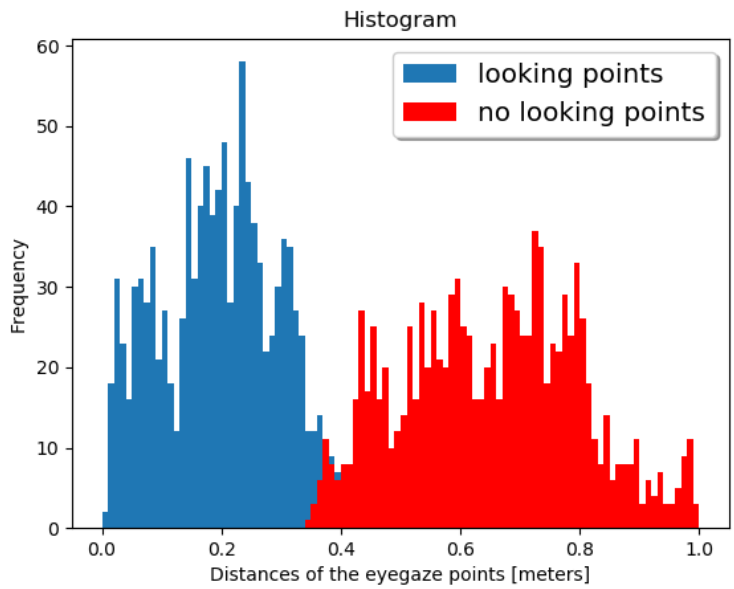


Figure 5.3: Histogram with the distances of the looking points and the no looking points for the eye gaze.

The distances in Figure 5.3 are in accordance with the discussion above and the final value obtained with the Bayes method for the boundary of the two classes is equal to 0.37 meters.

5.1.4 Experience 3 - Record data for head pose classifier and construct the classifier

Objectives

This experiment aimed to acquire data from people having a good and a bad head position to interact with the robot. This data was intended to train the head pose classifier and evaluate its performance.

Description

Two people, one male and one female, were asked to stand within 1 meter of the robot. Then they were asked to have a good head position as if they wanted to interact with the robot. They were also asked to do the opposite, have a bad position, that is as if they did not want to interact with the robot. Finally, the head pose classifier was trained and tested.

Results and discussion

As explained before, for the head pose module were used two machine learning algorithms, an SVM and a KNN. To train and test both, cross-validation was used as explained above in this chapter.

In Figure 5.4 are represented the accuracy results for the KNN and the different neighbors used, and in Figure 5.5 are represented the accuracy results for the SVM and the different values for γ and C used.

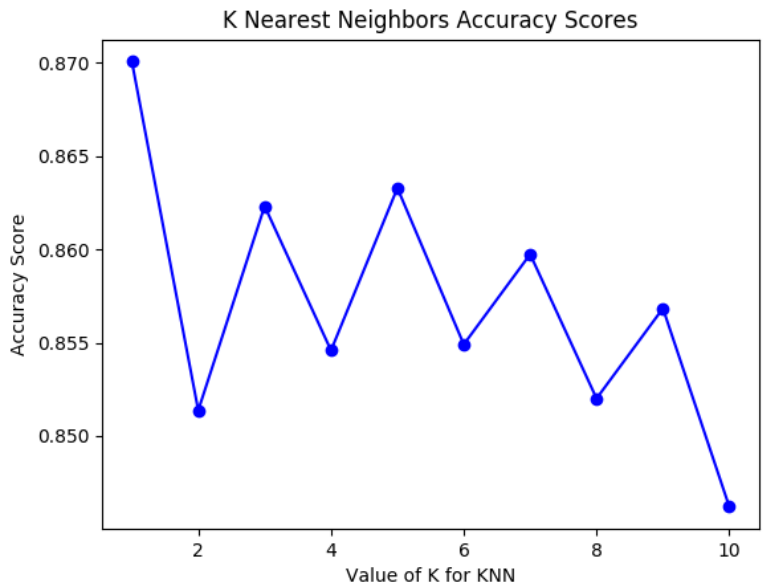


Figure 5.4: KNN cross validation results for the head pose experience.

Looking at the results of the two algorithms, the best accuracy results are given by the KNN with k=1 and SVM with γ = 10 and C = 1. Analyzing with more detail the accuracy score for these two situations, the score obtained for the SVM is a bit better so the classifier chosen for the head pose module was an SVM with γ = 10 and C = 1.

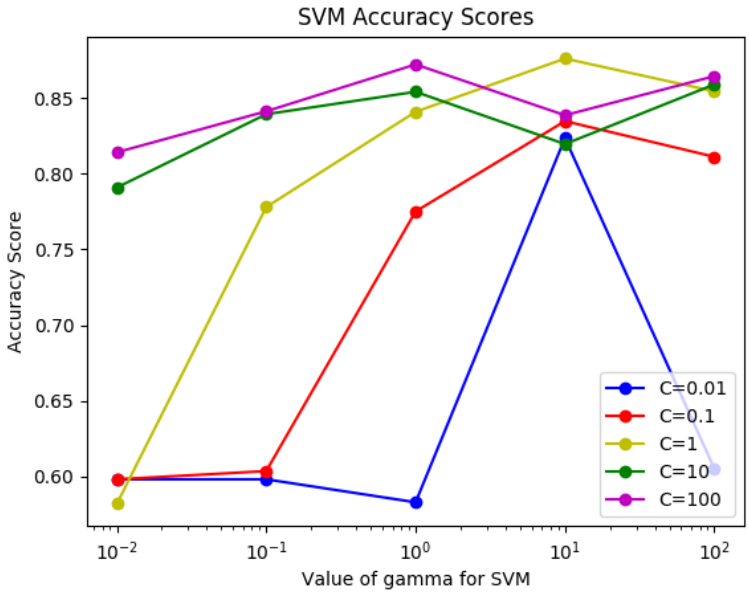


Figure 5.5: SVM cross validation results for the head pose experience.

5.1.5 Experience 4 - Record data for posture classifier and construct the classifier

Objectives

Like the experiences above, this one aimed to acquire data from people having a good and a wrong posture to interact with the robot. This data was intended to train the posture classifier and evaluate its performance.

Description

Two people, one male and one female, were asked to stand within 1 meter of the robot. After, they were asked to have a good posture as if they wanted to interact with the robot. They were also asked to do the opposite and have a wrong posture, that is as if they didn't want to interact with the robot. Were explained to the participants that proper or wrong posture, in this case, is associated with trunk rotation and how they presented the front of the body to the robot. In the end, the posture classifier was trained and tested.

Results and discussion

As was done for the head position module, the posture module also studied two machine learning algorithms, KNN and SVM. In Figure 5.6 are represented the accuracy results for the KNN and the different neighbors used and in Figure 5.7 are represented the accuracy results for the SVM and the different values for γ and C used.

In the Figures above it can be seen that the results for the accuracy of the algorithms are quite good, with results above 95 %. These results are mostly due to the fact that only shoulder-distance is used as

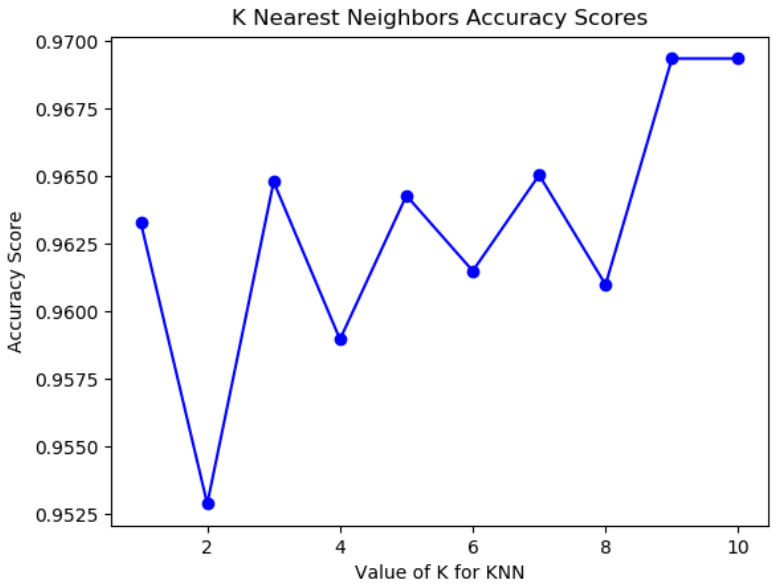


Figure 5.6: KNN cross validation results for the posture experience.

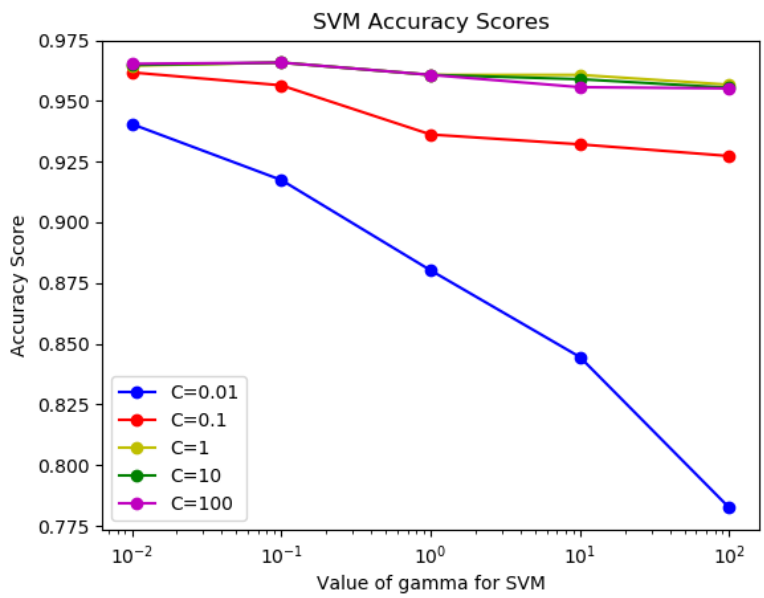


Figure 5.7: SVM cross validation results for the posture experience.

an input feature in these algorithms, and it is easy to find a distance that allows one classifier to have a boundary value when a person's posture is correct or wrong to interact with the robot, knowing that this posture is dependent on trunk rotation. Being the values easy to classify, for the SVM the results when C is increasing are very similar to the different values of γ.

Analyzing the accuracy scores of both algorithms, it was possible to decide that the best result is for the KNN with k = 9.

5.1.6 Experience 5 - Record data for final intent classifier and construct the classifier

Objectives

This experiment aimed to acquire data from people having an intention to interact without using gestures or speak to interact, so only be considered a potential intention of interaction. The experience would be able to detect people who were able to interact with the robot at that time. This data was intended to train the final intent classifier and evaluate its performance.

Description

Two people, one male and one female, were asked to stand within 1 meter of the robot. After, they were asked to be a potential intention of interaction, as if they wanted to interact with the robot. They were also asked to do the opposite as if they didn't want to interact with, the robot. For this, it was explained to the people that a potential intention is when the human is facing the robot and is in a posture that is able to start a communication with the robot through gestures. In the end, the final intent was trained and tested.

Results and discussion

With the costs obtained through the questionnaire made and the models for the eye gaze, head pose, and posture classifiers, all the information to the potential intent detector was acquired. Using (3.27) the final probability for this detector was obtained and used to create a histogram of probabilities for a situation when the person is a potential interacting agent or when is not. This histogram was made to obtain the value for the boundary that splits these two classes. The idea was the same as in the eye gaze module.

In Figure 5.8 is represented the histogram explained above.

Looking at the Figure 5.8 it can be seen that the results, although with some exceptions, present the expected, that is, that the probabilities of potential intent to interact were close to 1 and that the probabilities of non-potential intent to interact were close to zero. The exceptions that can be seen are the lack of results above 0.9 and quite a few non-potential intent results close to 0.4. The lack of values above 0.9 can be explained by the difficulty in obtaining a very high probability due to the fact that 3 modalities are used and the 3 do not always define very well the potential intent to interact of the person, as in the eye gaze module, which as already discussed, a person who looks at the robot in order to interact with it looks around the camera (distances are short) and not directly at it.

Regarding the high presence of values close to 0.4 for non-potential intent, this is due to the fact that both the test subjects present situations where they are in a correct posture but are not looking at the robot, however this direction of the gaze is a little distant from the robot which leads to the probability of the potential intent detector to these values obtained.

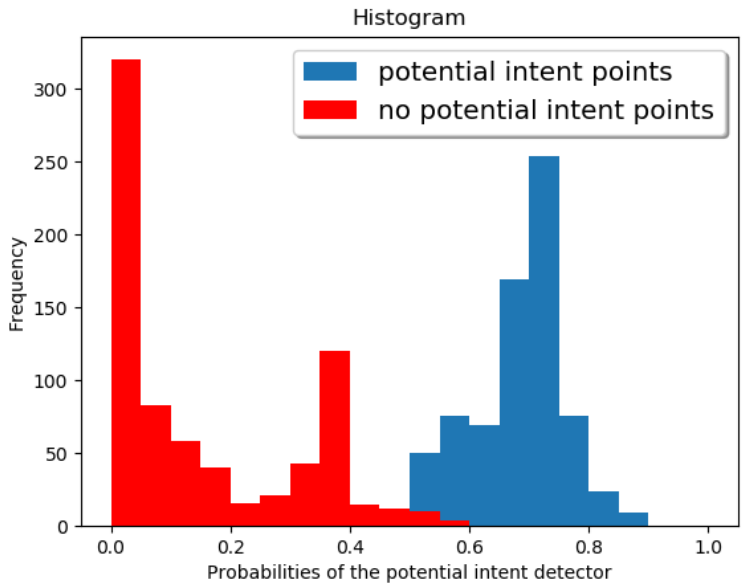


Figure 5.8: Histogram of the fused classifier scores (11) for the positive and negative examples of potential interaction intention.

In the end, the Bayes risk minimization method was implemented and the value for the boundary that enables the construction of the potential intent classifier was obtained. This value is equal to 0.45.

5.1.7 Experience 6 - Record data for gesture segments classifier and construct the classifier

Objectives

The goal of this experiment was to acquire data from people doing the gestures defined a-priori in this thesis for the gestures detector. With this data, it was intended to train the gesture segments classifier and evaluate its performance. Also, this data was intended to help construct the data-set to train the HMMs for the gestures.

Description

Four people, two male and two female, were asked to stand within 1 meter of the robot. After that, they were asked to do the gestures defined for the gestures detector. Each person was asked to do six times each gesture. This information could give the angles that identify each segment. Finally, the gesture segments classifier was trained and tested and these classified gestures were used to help construct the data to train the HMMs.

For the segments of the gestures, classes were created and in table 5.1 are demonstrated the movement that each class represents. These classes/labels were used for the classification of the segments in the classifiers studied.

For the gestures detector in the experiments performed the value used in the accelerations filter

Movement	Label
Lower any arm	0
Raise the right arm	1
Raise the left arm	2
Move the right arm part between the wrist and the elbow to the left	4
Move the right arm part between the wrist and the elbow to the right	5
Move the left arm part between the wrist and the elbow to the left	6
Move the left arm part between the wrist and the elbow to the right	7
Put the hands on the chest	8
Lower the trunk forward	9
Lift the trunk up	10

Table 5.1: Table for labeling the possible segments in gestures used in the gesture detector.

(allows the calculation of acceleration when the points of the same joint are far in two time-sequential points) is 30 pixels for every joint except the neck joint that is 20 pixels. This difference in the neck joint is to be possible to detect the angles for the bowing gesture.

Also, the window used to smooth the accelerations signal is the bartlett window represented in Figure 3.7.

Results and discussion

The results obtained with the segmentation of the gestures performed by each individual were used to train and test a KNN and SVM classifiers, so it was possible to study both and evaluate which one has the best performance. These results also were used as a possible sequence of segments to train the HMMs in the next experience that was done.

In Figure 5.9 are the accuracy results for the KNN and the different neighbors used and in Figure 5.10 are represented the accuracy results for the SVM and the different values for γ and C used.

Looking at the Figures mentioned above, it is found that by increasing the value of k-neighbors in KNN and gamma in SVM, the accuracy scores decrease, representing these values that the points that represent the classes used for each possible segment are close and both classifiers decrease a lot their performance by increasing these parametric values.

Analyzing both results obtained by both algorithms, it is possible to state that the best accuracy score is obtained for the KNN with k = 2 (see Figure 5.9). TThus, this is the classifier selected to incorporate in the final interaction intent detector.

5.1.8 Experience 7 - Record data for gesture sequence classifier, construct the classifier and evaluate

Objectives

This experiment aimed to acquire data from people about the possible segment's sequence for each gesture. Also, the experiment aimed to obtain data from people doing the gestures defined for the

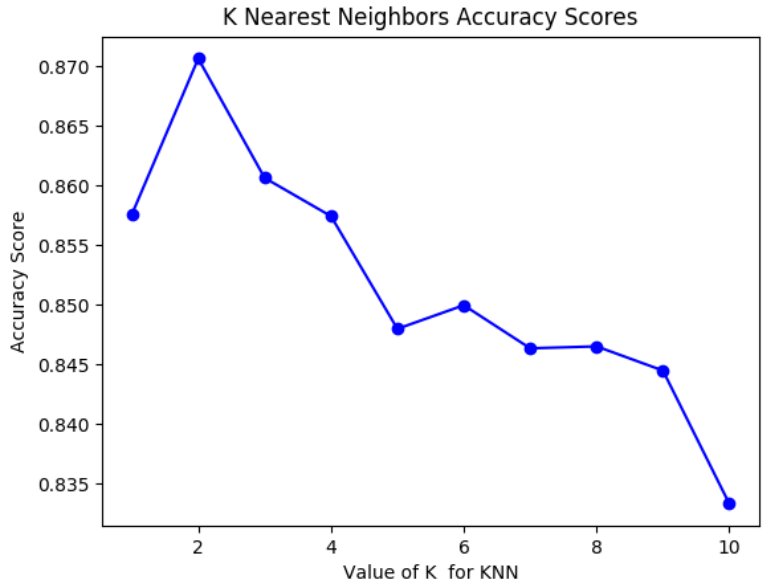


Figure 5.9: KNN cross validation results for the gestures segments experience.

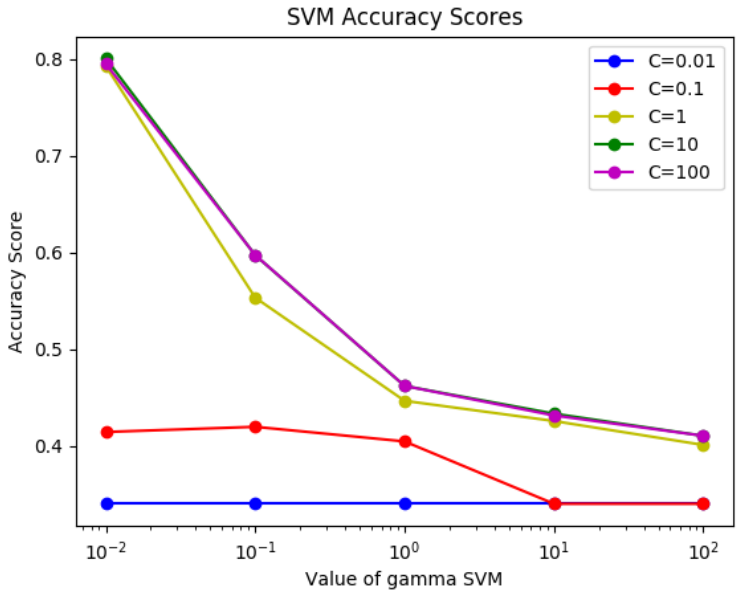


Figure 5.10: SVM cross validation results for the gestures segments experience.

gestures detector. With this data, it was intended to test the gesture sequence classifier and evaluate its performance.

Description

In the first part of the experiment six people, three male and three female, were asked to give ten possible sequences for each gesture, being the possible segments the ones presented in table 5.1. If the questioned person does not know more possible sequences, he or she can repeat the sequences that he or she thinks are most likely to happen for that particular gesture.

In the second part of the experiment, four people, two male and two female, were asked to stand within 1 meter of the robot. After that, they were asked to do the gestures defined for the gestures detector. Each person was asked to do five times each gesture. Finally, the gesture sequence classifier was tested with these gestures recorded.

Results and discussion

For the first part of the experience, the results were combined with the results obtained by the gestures segmentation made in experience 6. The combined results were used to train the 6 HMMs that represent the possible sequences for each gesture. The HMMs were trained with possible sequences of the known segments for the gestures.

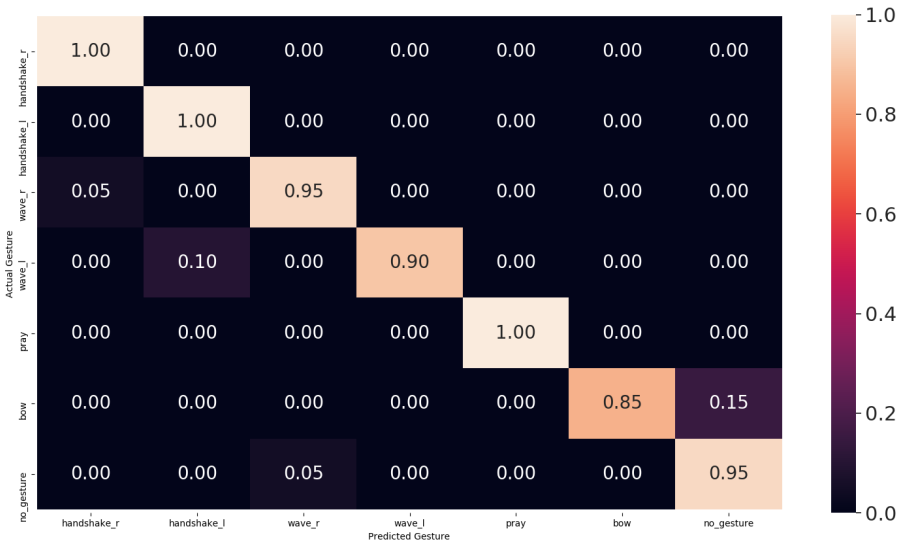


Figure 5.11: Confusion matrix for the dynamic gestures detector.

Having the models for the 6 gestures, these models were tested to evaluate the performance of the gestures detector. To evaluate this detector, as said above, 4 people made each gesture 5 times and those gestures whore segmented and classified. In Figure 5.11 is represented the confusion matrix obtained after analyzing the predict gestures by the gestures detector and the actual gestures made by the persons. In this Figure, the letter after the word ”handshake” or ”wave” represents the arm that does this gesture, being ”r” for right and ”l” for left.

Looking at Figure 5.11 is possible to conclude that the gestures detector for this experience had a good performance. Almost all gestures were predicted correctly, as the situations when no gestures were made. The gesture with less accuracy in its prediction was the bow, being the main problem the detection of this movement when the bow made by the human is not too forward. For both wave gestures, sometimes the gesture is too fast and gives results that can be confused with the handshake gestures like is represented in the confusion matrix.

5.1.9 Experience 8 - Final end-user validation

Objectives

The final experience aimed to validate and evaluate the system built. This experience aimed to see the results given by the completed system when a person wants to interact or doesn't want to interact.

Description

31 persons were asked to stand within 1 meter of the robot and interact with the robot knowing that the robot only knows 6 gestures. The gestures were told to these persons. After the interaction, was asked to each person to not interact with the robot at the same position that they interacted.

For this final experiment, the implementation referred to in this thesis was used. However, due to problems related to the robot, such as being unable to access it correctly or not being able to send image information to the server due to difficulties in the robot's Android operating system, an RGB camera was placed at the robot's camera height. This camera was connected to a computer, allowing the image to be sent to the server and to receive the information sent by the server. In conclusion, in this experiment, although without the robot, the entire implementation described in this work was tested with the exception of robot behavior control,i.e. people during the experiment did not receive an interaction if the robot detected an interaction made by them. The interaction processing and gestures,i.e. what the robot reads from the person, were displayed on the computer display during the experiment, allowing conclusions to be obtained about this tool.

Results and discussion

The results for the intention of interaction detector in this experiment are presented in 5.12. In table 5.2 is possible to visualize the number of times each gesture has been made for the 31 persons.

Gesture	Number of persons
Handshake right arm	9
Handshake left arm	2
Waving right arm	8
Waving left arm	4
Bow	4
Pray position	4

Table 5.2: Table for the number of persons in the experience that made each gesture.

Looking at this Figure is possible to infer that when a person doesn't want to interact, the results are very good because the robot detects no interaction with an accuracy of 96%. In these cases, the potential intent detector, detected that there were people able to interact as they just stared at the robot and have a good posture, however, as no gesture was performed, the interaction intent detector did not detect that this intention existed, as would be expected of a human who just pays attention to the robot or to what it is doing.

For cases where people interacted with the robot, almost all occasions were detected as interaction, with only a few not, having the tool, in this case, an accuracy of 87%. These bad identifications were because one person was trying to interact at a distance of more than 1 meter, leading to a large error in the potential intent detector and gestures that were not detected even though the persons were detected by the robot as being able to interact and made a gesture.

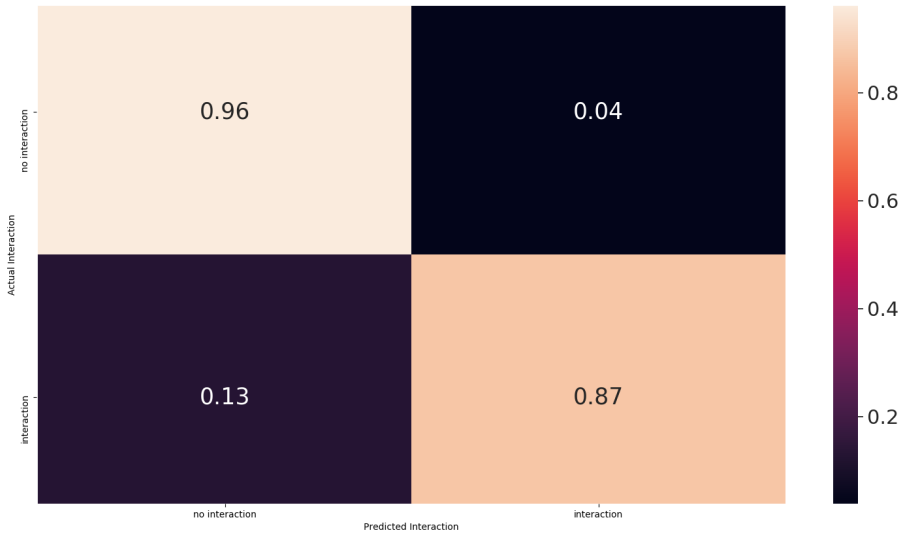


Figure 5.12: Confusion matrix for the intention of interaction detector.

The tool, having detected interaction, correctly identified gestures 78.6% of the times. So the final accuracy for the gestures predicted correctly was equal to 68.4%. The main reasons of failures are due to a person bowing while praying, poorly identified handshakes due to the high velocity of gesture and misidentified bows. The problem with the high velocity of gestures is that the tool used to analyze the body joints, Openpose, works only at 10 FPS which is low for dynamic gestures. The number of interactions detected was 28 and from these 28 interactions, only 6 gestures were badly identified.

In Figure 5.13 are represented 4 cases studied. In (a) the person was looking to the camera and had a good head and body position, so the system detected a potential intent of interaction. In (d) all the above weren't true, the person didn't want to interact and the system detected a no potential intent. The cases (b) and (c) are particular. In (b) the person wasn't looking to the camera but had a good head pose and a good posture. Although the person wasn't looking to the camera, since he was far away, the eye gaze module used the direction of the person's face, which led the system to detect a potential agent that wanted to interact. For last, in case (c), the person had a bad posture and a not so good head position and was looking to the camera. The system didn't detect a potential intent, although the person was looking to the camera.

(a) Looking to the camera and good posture to interact.

(b) Not looking to the camera, good head position and good posture to interact.

(c) Looking to the camera and bad posture to interact.

(d) Not looking to the camera and bad posture to interact.

Figure 5.13: Developed system detecting potential intention of interaction. The red label represents no potential intention detected and the green label it's the opposite.

Chapter 6

Conclusions

6.1 Achievements

Through this work, it was possible to study and develop a tool capable of detecting the intention of human interaction with a robot, using only information from an image obtained with an RGB camera.

This detector was divided into two modules and it was possible to evaluate the performance of each of these modules individually, being possible, in the end, to join this information. This combination enabled to obtain values that allowed the evaluation of the intention of interaction detector as a whole.

Regarding the potential intent detector, it was possible to verify the importance of the eye gaze and posture so that the robot can identify a certain intention made by the human. It was also possible to verify that the implemented methods allowed the construction of good classifiers and that the multimodal system allows giving robustness and accuracy to the system, because the eye gaze sometimes presents some detection failures, due to the fact that at long distances the associated values to this modality present some error.

As for the dynamic gesture detector studied and implemented, the results were good and allow us to state that the use of the presented methodology leads to good interactions between the robot and the human. Through this detector, it was possible to perform the segmentation and classification of gestures with relatively low errors, and only for certain gestures, such as the bow, which are associated with proximity gestures to the camera, the results were not so good.

It was also possible to prove that for the head position case, the best classifier to use among those studied is an SVM with C = 1 and γ = 10, while for the posture case, the best classifier to use is a KNN with k = 9. In addition to these two, it is also concluded that to classify the segments of gestures, a KNN with k = 2 provided the best results.

Combining those last two detectors mentioned, the intention of interaction detector was tested by 31 people and achieved an accuracy of 96% for cases where people did not interact and 87% for cases where people interacted. In cases where people interacted the accuracy of gesture detection was 78.6%. These results allow us to conclude that the detector of intent to interact is a good detector for cases when the person is at approximately 1 meter from the robot. However, has a slightly high error in

gesture detection. This detector, being 2D and using a tool (Openpose) that only works at 10 FPS, leads to some errors in the detection of gestures made very quickly or in cases where the bow gesture is performed but with low intensity, not allowing a very correct reading of the λ value.

6.2 Future Work

In this work, an intention of interaction detector was constructed and divided into two modules already explained: the potential intent detector and the gestures detector.

As improvements to these two realized modules, it should be noted that for the potential intent detector module, the use of other modalities such as speech would be interesting to be analyzed, or to be analyzed other body postures made by the individual who wants to interact with the robot, so that the posture module is more robust to different positions that the individual can do. For the gesture detector, it would be interesting to analyze another type of gesture and combine this dynamic gesture detector with a solution that would allow analyzing the gestures made by the individual's hand. There are gestures that are dynamically the same but have a different connotation, such as a handshake and an "ok" made through a person's hand when he/she directs this information to the robot. Therefore, it would be interesting to implement a tool that, after analyzing the dynamic gesture made by the 25 joints in the human body, would be able to make a cropping of the image of the intervener's hands and be able to analyze the gesture made with the combination of hand information with dynamic information, improving the performance of the robot's ability to interact with the human.

Another aspect to improve, is that the work was developed for the robot to interact with only one person, that is, the robot analyzes the information of one person and interacts only with it even if there are more people who want to interact. Thus, it is important to improve the tool created so that it's possible to analyze and interact with everyone present in the image acquired by the robot's RGB camera.

Using only one RGB camera has its disadvantages, and as explained throughout this thesis, for the different modules, the distance from person to the robot is an important aspect. That said, the solution is not just about using an RGB-D camera. In this thesis the assumption is made that the individual who wants to interact with the robot is 1 meter away, however, there are algorithms that, through calculated distances, such as the person's height, can calculate the distance from the camera. The use of algorithms and methods that allow this action would be an improvement to this work so that the results obtained for the different modules are more in accordance with reality.

Finally, another interesting aspect that needs to be improved is the speed of information given by the gesture analysis tool. Openpose, the tool used, has a frame rate of 10 seconds, which means that gestures made quickly are not well identified and segmented. It would be important to use tools that give the same information that Openpose gives, however having FPS higher than the Openpose, improving the detection and segmentation of the gestures.

The code for the intention of interaction detector implemented and the data-sets used to train and test the classifiers, are available at `https://github.com/soaresafonso/Intention_Interaction_Detector.git`.

Bibliography

[1] D. Feil-Seifer and M. Mataric. Human-robot interaction. In *Encyclopedia of Complexity and Systems Science*, pages 4643–4659. 2009.

[2] K. Dautenhahn. Methodology themes of human-robot interaction: A growing research field. *International Journal of Advanced Robotic Systems*, 4:103–108, 2007.

[3] B. Ogden and K. Dautenhahn. Robotic etiquette: Structured interaction in humans and robots. Technical report, University of Hertfordshire, College Lane, Hatfield, Herts AL10 9AB, 2000.

[4] C. Breazeal, C. Kidd, A. Thomaz, G. Hoffman, and M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 383–388, 2005.

[5] J. Sanghvi, G. Castellano, I. Leite, A. Pereira, P. W. McOwan, and A. Paiva. Automatic analysis of affective postures and body motion to detect engagement with a game companion. *In 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2011.

[6] J. M. Salvado. Detection of interaction intention and engagement in hri using joint gaze and gesture analysis. Master dissertation, Instituto Superior Técnico, Portugal, Lisbon, November 2017.

[7] D. Losey and M. O'Malley. Enabling robots to infer how end-users teach and learn through human-robot interaction. *IEEE Robotics and Automation Letters*, 4:1956–1963, 2019.

[8] C. Jiang, Z. Ni, Y. Guo, and H. He. Learning human–robot interaction for robot-assisted pedestrian flow optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:797–813, 2019.

[9] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[10] X. Gao, M. Zheng, and M. Q.-H. Meng. Humanoid robot locomotion control by posture recognition for human-robot interaction. *In IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015.

[11] A. Tapus, A. Bandera, R. Vazquez-Martin, and L. V. Calderita. Perceiving the person and their interactions with the others for social robotics – a review. *Pattern Recognition Letters*, 118:3–13, 2019.

[12] C. Heinze. Modeling intention recognition for intelligent agent systems. Doctoral dissertation, University of Melbourne, Australia, 2003.

[13] S. Trick, D. Koert, J. Peters, and C. Rothkopf. Multimodal uncertainty reduction for intention recognition in human-robot interaction. *In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2019.

[14] D. W. Hansen and Q. Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE, Transactions on Pattern Analysis and Machine Intelligence*, 32, 2010.

[15] C. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, and C. Parlitz. A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction. pages 371–377. 2005.

[16] R. Stiefelhagen, M. Finke, J. Yang, and A. Waibel. From gaze to focus of attention. proceedings of workshop on perceptual user interfaces: Pui. *Proceedings of Workshop on Perceptual User Interfaces: PUI 98*, 1614:25–30, 1999.

[17] T. Baltrušaitis, A. Zadeh, Y. Lim, and L.-P. Morency. Openface 2.0: Facial behavior analysis toolkit. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2018.

[18] E. Wood, T. Baltrušaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[19] A. Gaschler, J. S, M. Giuliani, K. Huth, J. de Ruiter, and A. Knoll. Social behavior recognition using body posture and head pose for human-robot interaction. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2128–2133, 2012.

[20] M. Nakai, Y. Tsunoda, H. Hayashi, and H. Murakoshi. Prediction of basketball free throw shooting by openpose. *In International Skill Science 2018, At Tokyo Japan*, 2018.

[21] J. Lee, C. Hsieh, and T. Lin. A preliminary study of using kinect-based physical rehabilitation system to perform tai chi exercises with fls evaluation. *Neuropsychiatry*, 8:165–175, 2018.

[22] K. M. Sagayam and D. J. Hemanth. A probabilistic model for state sequence analysis in hidden markov model for hand gesture recognition. *Computational Intelligence*, 35:59–81, 2019.

[23] H. Liu and L. Wang. Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics*, 68:355–367, 2018.

[24] S. Qiao, Y. Wang, and J. Li. Real-time human gesture grading based on openpose. *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017.

[25] W. Benkaouar and D. Vaufreydaz. Multi-sensors engagement detection with a robot companion in a home environment multi-sensors engagement detection with a robot companion in a home

environment. *Workshop on Assistance and Service robotics in a human environment. In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 45–52, 2012.

[26] S. Kim, Z. Yu, J. Kim, A. Ojha, and M. Lee. Human-robot interaction using intention recognition. *In Proceedings of the 3rd International Conference on Human-Agent Interaction*, pages 299–302, 2015.

[27] F. Timm and E. Barth. Accurate eye centre localisation by means of gradients. *Visapp*, pages 125–130, 2011.

[28] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[29] T. Baltruaitis, P. Robinson, and L. P. Morency. Constrained local neural fields for robust facial landmark detection in the wild. *Proceedings of the IEEE International Conference on Computer Vision*, page 354–361, 2013.

[30] D. Wu, H. Zhu, and E. Zhou. A bayesian risk approach to data-driven stochastic optimization: Formulations and asymptotics. *SIAM Journal on Optimization*, 28:1588–1612, 2018.

[31] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. Knn model-based approach in classification. *In CoopIS/DOA/ODBASE*, 2003.

[32] J. C. Platta. Fast training of support vector machines using sequential minimal optimization. In *Advances in kernel methods*, pages 185–208. MIT Press Cambridge, feb 1999.

[33] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[34] M. Lee, D. Siewiorek, A. Smailagic, A. Bernardino, and S. B. Badia. Learning to assess the quality of stroke rehabilitation exercises. *24th International Conference on Intelligent User Interfaces*, Mar. 2019.

[35] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9:611–629, Aug. 2018.

[36] C. Creider. Hand and mind: What gestures reveal about thought. *Journal of Linguistic Anthropology*, 4:81–82, 1994.

[37] H. Drewes and A. Schmidt. Interacting with the computer using gaze gestures. *Proc. of the Int. Conf. on Human-computer interaction'07*, page 475–488, 2007.

[38] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 37:311–324, 2007.

[39] K. Dautenhahn, C. Nehaniv, J. Kubacki, M. Haegele, and C. Parlitz. A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction. *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication*, page 371–377, 2005.

[40] A. Wilkowski. A hmm-based system for real-time gesture recognition in movie sequences. *2008 Conference on Human System Interactions*, page 743–742, 2008.

[41] H. Badi, S. Kareem, and S. Husein. Feature extraction technique for static hand gesture recognition. *A. Chaudhary. Recent Trends in Hand Gesture Recognition*, 3:19–41, 2015.

[42] J. Yang and W. Choi. Smooth trajectory planning along bezier curve for mobile robots with velocity constraints. *International Journal of Control Automation*, 2013.

[43] A. Zelinsky. Learning opencv-computer vision with the opencv library. 2009.

[44] T. Baltrušaitis, M. Mahmoud, and P. Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection in facial expression recognition and analysis challenge. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2015.

[45] T. Baltrušaitis, P. Robinson, and L.-P. Morency. Constrained local neural fields for robust facial landmark detection in the wild. *IEEE Int. Conference on Computer Vision Workshops, 300 Faces in-the-Wild Challenge, 2013.*, 2013.

[46] A. Zadeh, T. Baltrušaitis, and L.-P. Morency. Convolutional experts constrained local model for facial landmark detection. *Computer Vision and Pattern Recognition Workshops*, 2017.

[47] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[48] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[49] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[50] K. Vishi and V. Mavroeidis. An evaluation of score level fusion approaches for fingerprint and finger-vein biometrics. *In the 10th Norwegian Information Security Conference (NISK 2017)*, 2017.

[51] I. Qihan technology co. In *Qihan Sanbot S1-B2 User Manual*.