UNIVERSITÀ DELLA CALABRIA

Dipartimento di Fisica

Scuola di Dottorato

“Archimede” in Scienze, Comunicazione e Tecnologie

Indirizzo:

Scienze e Tecnologie dei Sistemi Complessi

XXVIII CICLO

TITOLO TESI

A machine learning system for developing a Human-Robot interface for automatic facial emotions and hand gestures recognition

Settore Scientifico Disciplinare

MAT/07

Coordinatore: Ch.mo Prof. Pietro Pantano

Firma Pietro Pantano

Supervisore: Ch.mo Prof. Pietro Pantano

Firma Pietro Pantano

Dottorando: Dott. Leonardo F. Renteria B.

Firma

Abstract

Emotions are an essential part of people's lives because they help us to make decisions, communicate and somehow to understand each other. Additionally, facial emotion perception plays an important role in various fields of psychology, neuroscience, computational intelligence and robotics. In the last years, robots have stopped being used as simple machines dedicated to carry out repetitive, difficult and dangerous jobs. Thanks to rapid technological advances and the emergence of more powerful computers, robotic applications have extended beyond the commercial and industrial scope becoming more popular in the fields of education and medicine. With the development of humanoid and interactive robots such as ASIMO from Honda, AIBO from Sony and currently NAO from Aldebaran, it has emerged a new line of research in robotics. Worldwide, research groups have increased their work on topics such as bipedal locomotion, manipulation, audio-visual systems, human-robot interaction, adaptive control, and learning tasks. This work shows the development of a Machine Learning system to equip the robot NAO with the capacity not only to recognize the people but also to recognize people’s facial expressions and hand gestures. An analysis of the facial symmetry and hand contour geometry was carried out in order to improve the facial expressions and hand gestures recognition rate. In addition, the robot was programmed to express emotions through movement, sound and colors, making the interaction with people more friendly and eye-catching.

Le emozioni rappresentano una parte essenziale nella vita degli individui, in quanto necessarie nella scelta delle decisioni, nella comunicazione e nella comprensione.

Importante è anche la percezione delle espressioni facciali che gioca un ruolo rilevante nei vari ambiti della psicologia, della neuroscienza, dell'intelligenza computazionale e della robotica. Negli ultimi anni, i robot hanno smesso di essere semplici macchine dedicate allo svolgimento di compiti difficili, ripetitivi e pericolosi. Infatti, grazie all'avanzare delle tecnologie e alla nascita di strumenti sempre più potenti, il campo di applicazione della robotica si è esteso, passando dall'ambito commerciale e industriale anche a quello dell'educazione e della medicina. Con lo sviluppo dei robot umanoidi e interattivi quali: ASIMO di Honda, AIBO di Sony e di recente NAO di Aldebaran è emersa una nuova linea di ricerca nel campo della robotica. A livello mondiale si è, infatti, registrato un aumento di gruppi di ricerca che lavorano su temi quali: la locomozione bipede, la manipolazione, i sistemi audio-visivi, l'interazione human-robot, il controllo adattivo e le attività di apprendimento. In questo lavoro si presenta lo sviluppo di un sistema di Machine Learning in cui ad un robot NAO è stata assegnata la capacità di riconoscimento di una persona, di riconoscimento delle espressioni facciali e di riconoscimento dei gesti della mano. Successivamente è stata condotta un'analisi riguardo la simmetria facciale al fine di migliorare la percentuale di riconoscimento delle espressioni facciali e il contorno della mano al fine di migliorare la percentuale di riconoscimento dei gesti. Infine il robot è stato programmato per esprimere le emozioni tramite i movimenti, i suoni e i colori rendendo l'interazione piacevole e attrattiva.

Dedication

To my mother who is an important inspiration for me and to the memory of my father Fico and my brothers Gina e Pato.

Acknowledgements

First, I would like to express my sincere gratitude to University of Calabria and to the professor Galileo Violini, who in capacity of delegate of the rector for international relations, privided me the opportunity to do my PhD studies in this university.

Thanks to Evolutionary Systems Group for letting me to develop my research project here, especially to my supervisor prof. Pietro Pantano for his valuable guidance and prof. Eleonora Bilotta for her assistance and suggestions. To my laboratory colleagues for their collaboration, in particular to Francesco Chiaravalloti for his valuable help during my graduate education and Adolfo Adamo for his contribution performing the facial expressions used in this project.

I am grateful to the Cascina UniCal, especially the ladies Anna Salituro, Rosalia Barillaro, Rosaria Calabrò and Rosetta Sciommarella for to their warm welcoming and charming hospitality which made my stay and studies in Italy more pleasant. To all my friends, especially to members of the C8 GIOVEDÌ H20 PROCOSENZA soccer team for keeping me entertained every Thursday, in particular to Salvatore La Porta for giving me the opportunity to do in Italy what I love; playing soccer.

Thanks to my family members for supporting, feeding and loving me throughout my life, in particular to my brother Jorge for his time and effort in checking this document. Finally, my gratitude to Margarita Mayacela, the greatest person I know, whose love, encouragement and support have made everything worth doing.

Contents

List of figures

List of tables

Chapter 1

Introduction

Emotions are essential in people's lives [1], these can influence attention, memory and logical reasoning; In addition, emotions have a strong impact on daily routines and social interactions [2]. Emotions help make decisions, communicate and somehow understand each other. Paul Ekman is a pioneer in the recognition of emotions and facial expressions and defined the universality of six basic emotions: anger, happiness, fear, surprise, disgust and sadness [3]. In the field of the psychology, facial expression is only one of the relevant sources of information of the emotions [4].

On the other hand, facial emotion perception plays an important role in several areas of neuroscience, computational intelligence and robotics. Facial expressions do not necessarily allow us to perceive a person's emotions or mood [5]; However, constitute one of the strongest indicators of emotions [2]. The fields of application of facial expressions recognition are so extensive and impressive as: security, animation and entertainment, neuromarketing, medical and psychological treatments and of course in interactive robotics and engineering of artificial intelligence.

Human-Robot Interaction (HRI) is a discipline that studies relationship between robots and humans and the ways they influence each other [6]. Up until recently, robots have been used in many fields as programmable machines for execution of preprogrammed tasks such as painting, welding, simple assembly task, etc. [7]. However, the use of robots in other scopes such as domestic jobs, health, academic and so on are driving the development of interactive robots to engage with individuals as a partners, instead of just a tool.

This projects aims to develop a machine learning system for automatic recognition of facial expressions and hand gestures for a human robot interface. The NAO Robot was chosen as the hardware platform; it is a programmable and autonomous humanoid robot, developed by Aldebaran Robotics [8].

This thesis comprises six chapters; first, a theoretical introduction about humanoid robots, the technical characteristics of the NAO Robot, and the programming tools are discussed. Chapter 2, briefly describes some of the most relevant research groups dedicated to the development of humanoid robots. Chapter 3 details the most important features of the NAO Robot technology and its available programming tools. It is also provided a short introduction on the use of the Choregraphe programming environment that is included within the NAO Robot features. A series of interactive applications (Chapter 4) were developed in Python and tested independently. Telecontrol (Teleoperations) was the first feature provided to the NAO. The system was based on the WWW technology and allows to access and / or control the NAO from a remote place through the network. In addition, a classifying machine was developed to provide the NAO robot with the ability not only to detect but recognize faces, as well as facial expressions and hand gestures. Also, the NAO Robot was programmed to express emotions through movements, sounds and colors in the eyes. An analysis of the symmetry of the face and the geometry of the contours of the hands was carried out in order to improve the success rate of the recognition system of facial expressions and hand gestures respectively.

In the last section (Chapters 5), the applications developed were used within the Choregraphe programming environment through the "AlMemory" module available in the NAOqi framework that allows access to the robot variables remotely with an operation Simple writing / reading.

Finally, the NAO Robot was used in an interactive activity with students from several local institutions, the robot developed three specific tasks that consisted of a quiz of general Italian culture, recognition of facial gestures and counting of the fingers. This experiment was evaluated qualitatively.

Chapter 2

The humanoid robots

Usually, a robot is a physical or virtual machine able to perform autonomously or semi-autonomously a specific task. Robots are generally used in commercial or industrial areas to perform repetitive, mechanical and monotonous tasks. They are also used for dirty or dangerous jobs to humans. The extraordinary advances in the field of robotics driven by cheap electronic devices and its greater processing power, invite to fantasize a robotized future, however, compared with their counterparts in fiction, real robots are still significantly limited.

A humanoid robot is a robot that mimic body and movements of a human being. In general, it has a torso, a head, arms and legs; although, it may feature only some parts of the body, for example, from the waist up. Some humanoids are built to seem humans and sometimes, they can replicate the facial features such as eyes and mouth[9].

So far, research on humanoids robots has centered on motion, manipulation, locomotion stability and other basic tasks in robotics control [10]. Particularly the last two decades can be considered as a prolific period on humanoid robots development. Many improvements have been done on advanced materials, mechanical, electronic and computer technologies [11]. However; nowadays, research on humanoids robots is focusing on behavior, emotions and interaction with humans. In this regard, the design of humanoid robots requires the integration and coordination of diverse related areas as learning theory, control theory, artificial intelligence, mechatronics and even biomechanics and computational neuroscience [12].

With the development of humanoid and interactive robots such as Honda's ASIMO [13], Sony's AIBO [14] and Aldebaran's NAO [8], a new research direction in robotics has emerged [15]. Worldwide, research groups have increased working on issues such as bipedal locomotion, manipulation, audio-visual systems, human-robot interaction, adaptive control, and learning tasks [16].

Honda

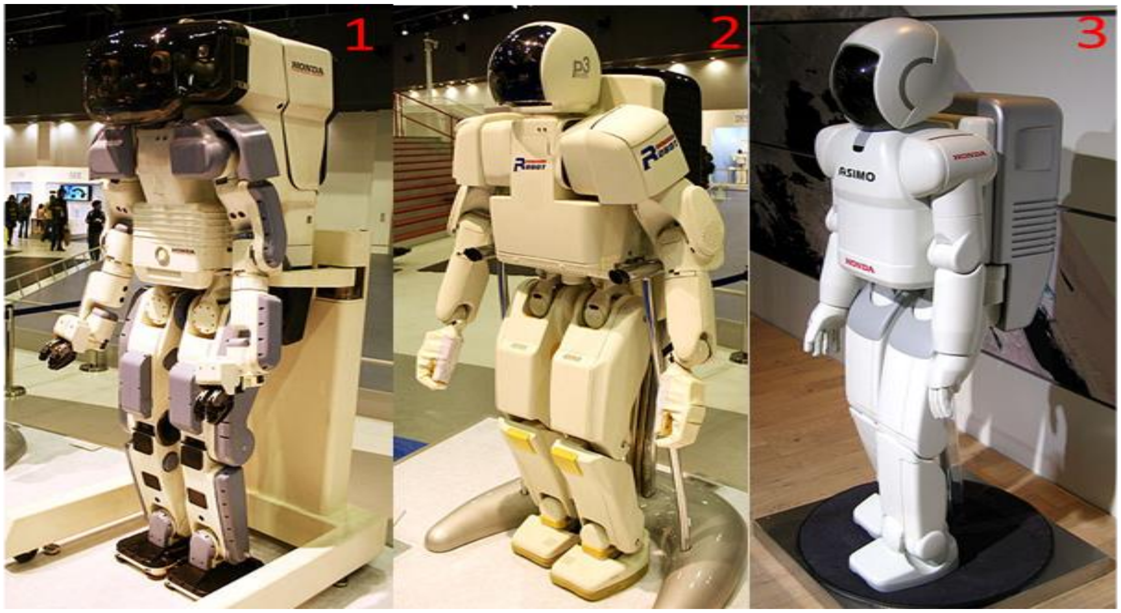


Figure 1. Honda's Robots. (1) P2, (2) P3 and (3) Asimo [17].

After several years of research, in 1996 Honda introduced P2 to the public, the first full-body humanoid robot (Figure 1, box 1). It was able to walk not only on flat floors, but it could also climb stairs. It was followed by P3 (Figure 1, box 2) in 1997 and by Asimo (Figure 1, box 3) in 2002 [16]. Asimo can walk like a human in any direction, climb stairs and recognize faces, voice and gesture commands, etc. The latest version of Asimo presented in 2011, can run up to 9 km/h and has a decision-making capability [17].

WABIAN-2

Researchers at Waseda University began investigating humanoid robots since 1966 [11]. Since then, they have developed a series of robots such as the WABOT-1 (Figure 2, box 1) in 1973, which is a full-scale human-like robot or the biped walking robot WABIAN (Figure 2, box 2) in 1997 [18]. They have recently developed a biped humanoid robot WABIAN-2R (Figure 2, box 3), a life-size humanoid able to walk with the average rate of a human; it is 150 cm tall and weighs 64 kg [19]. Unlike most robots that walk with bent knees, it is able to walk stretching it’s knees the same way as human does [20].

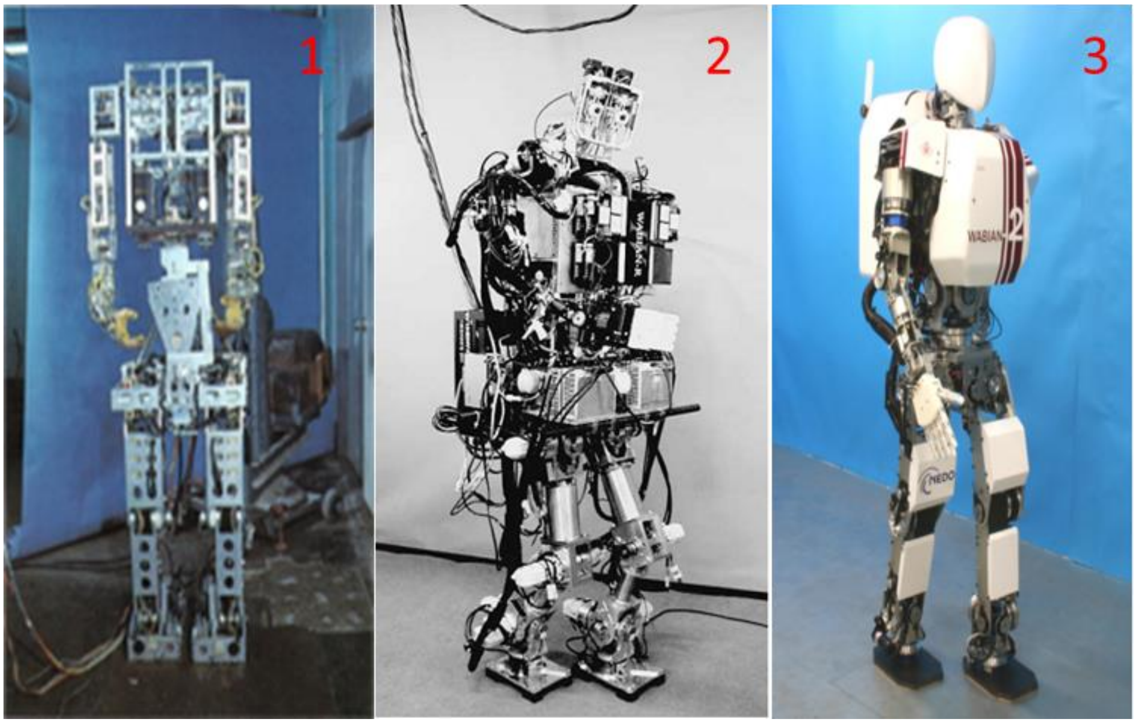


Figure 2. Waseda's Robots. (1) Wabot-1, (2) Wabian and (3) Wabian-2r [19].

iCub

The iCub (Figure 3) is part of the RobotCub project funded by European Union; it is a common platform for artificial cognitive systems research [21]. It was designed to maximize the number of Degrees of Freedom (DoF), especially in the upper part of the body [22]. The size of the iCub is similar to the stature of a 2.5-year-old child. It has 53 motors that move the head, arms & hands, waist, and legs. In addition to seeing and hearing, it is provided with the sense of proprioception (body configuration) and movement (using accelerometers and gyroscopes) [21].

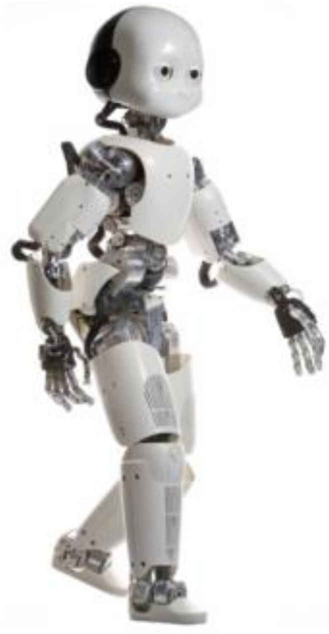


Figure 3. iCUB robot [23].

The iCub has the capability to carry out complex tasks such as crawling, solving complex 3D puzzles [24], or successfully shooting arrows [25]. It is able to express emotions, grabbing small objects, such as balls, plastic bottles, besides, it can avoid collision and self-collision within dynamic environment [26], [27].

HUBO

This is a walking humanoid robot released in 2005. It was developed by the Korea Advanced Institute of Science and Technology (KAIST) [28]. HUBO (Figure 4, box 1) has 41 DoF, stands 125 cm tall, and weighs 55 kg [29]. HUBO can only walk at 1.25 km/h [30]. The Hubo project aim was to develop a reliable platform that enables to perform tasks as navigation, dynamic walking, human interaction, visual and image recognition [28].

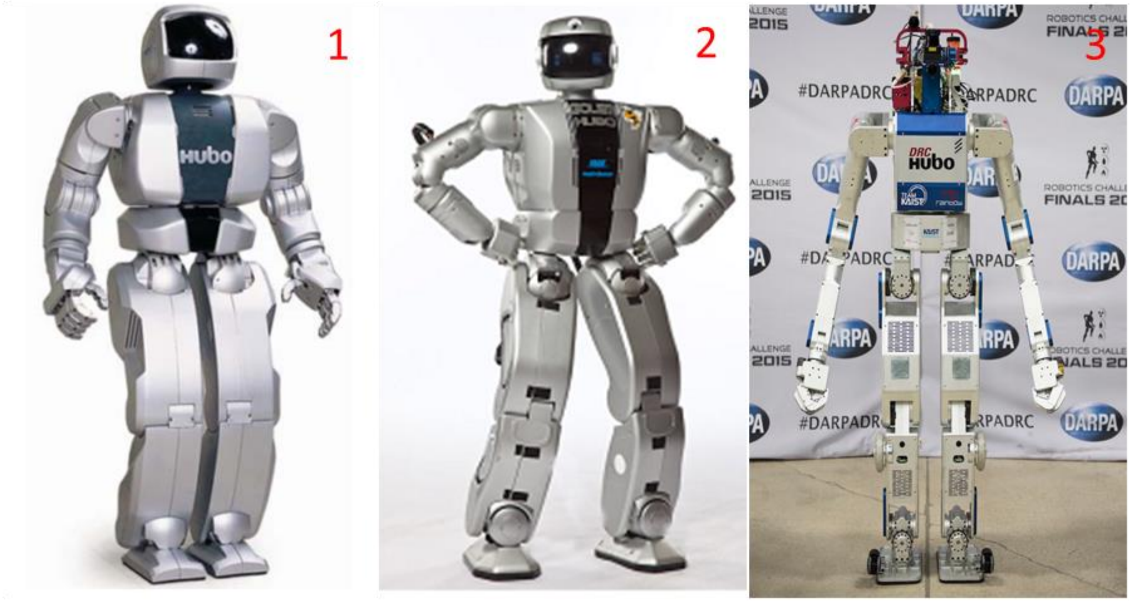


Figure 4. KAIST's Robots: (1) Hubo [31], *(2) Hubo 2+*[32] *and (3) DRC-HUBO* [33].

The latest version of HUBO is the DRC-HUBO (Figure 4, box 3) robot developed by KAIST/Rainbow, Inc. in 2013. It is 147 cm tall, weighs 52 kg and it has 32 DoF [34]. With its particular "transformer" ability to switch back and forth from a walking biped to a wheeled machine[33], it is the winner of the DARPA Robotics Challenge [35]. It is provided with longer and stronger aluminum limbs, becoming a taller and heavier robot than its predecessor, the HUBO 2+ (Figure 4, box 2) [34].

NAO

NAO is a programmable and autonomous humanoid robot, developed by Aldebaran Robotics, a robotics French company based in Paris (Figure 5). With 57 cm high and around 4.5 kg weight, this innovative robot is lightweight and compact and, due to its reduced weight, it is less hazardous and less susceptible to breakdown [36]. At present, NAO is widely used as a social companion, as a learning tool in classroom and for autism rehabilitation purposes [37]. Additionally, there are already successful applications of NAO humanoid robots in hospitals [38].

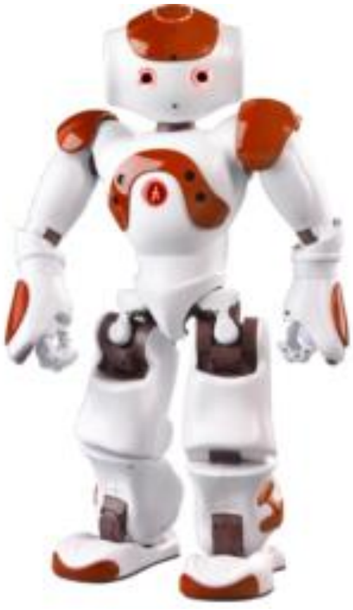


Figure 5. NAO Robot [8].

Pepper.

Softbank Mobile, in collaboration with Aldebaran Robotics, launched Pepper on June 5th, 2014 (Figure 6), the world’s first humanoid robot able to assist humans by reading and responding to human emotions [39]. Pepper is a humanoid robot that moves on a mobile holonomic base. It is equipped with three omnidirectional wheels [40], 2 DoF head, two 6 DoF arms, and a unique 3 DoF leg [41]. It is 137 cm height and weights 30 kg.

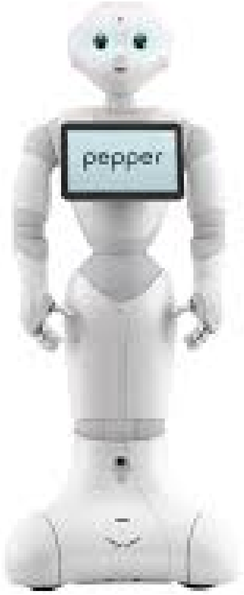


Figure 6. Pepper robot [39].

Pepper was developed to express emotions, it is able to communicate on a wide range of subjects with humans through its autonomous behavior, speech, and emotional recognition function abilities as well as its smooth motion-generation technology [42].

Chapter 3

NAO Technical Overview and Programming Tools.

The most popular research platforms in the field of robotics and artificial intelligence are robots with numerous DoF, such as, humanoid robots [43]. The NAO Robot, made by Aldebaran Robotics, is a small biped, full articulated, easy programmable and low cost product. It is widely used for humanoid research due to ease of use [44]. The Aldebaran Robotics main goal was to create a humanoid companion robot accessible to the public [45]. NAO is able to perform smooth walking gaits, the walking speed is about 0.6 km/h [36].

Different versions of the NAO platform have 14, 21 or 25 DoF, therefore, it can perform several complex moves like walking, kicking a ball, standing up, etc. All NAO Academic versions offer: inertial measurement, accelerometer, gyrometer and ultrasonic sensors that provide stability and positioning in space (Table 1)[46]. It is equipped with two 640x480 VGA cameras, microphones and loudspeakers, besides, it is provided with infrared, force, contact and touch sensors [46].

The NAO H25 (V4) carries a fully capable computer on-board with an ATOM Z530 1.6 GHz CPU, 1GB RAM, and 2 GB, 8 GB Micro SDHC flash memory running an Embedded Linux distribution. It communicates with remote computers via an IEEE 802.11g wireless or a wired Ethernet link and it is powered by a 21.6 V / 2.25 Ah Lithium-Ion battery which provides about 60 minutes of continuous operation and [8].

Table 1. NAO's main features.

Name	Chain
Degrees of freedom	25
Videocamera	2
Gyrometer	2
Three axis accelerometer	3
Magnetic rotary encoder (MRE)	36
Loudspeakers	2
Microphones	4
Ultrasonic sensors (emitter/receiver)	2
Force Sensitive Resistors	8
Infrared sensor (emitter/receiver)	2
Tactile sensors	9
Bumpers	5

Joint Degrees of Freedom (DoF)

NAO H25 has 25 DoF (Figure 7); 2 in the head, 5 in each arm, 5 in each leg, 1 in each hand and 1 in the pelvis (two pelvis joints are coupled together on one servo so it cannot move independently) [47]. The joins position is controlled using closed-loop PID controllers and encoders. All joints are organized into five kinematic chains (Table 2).

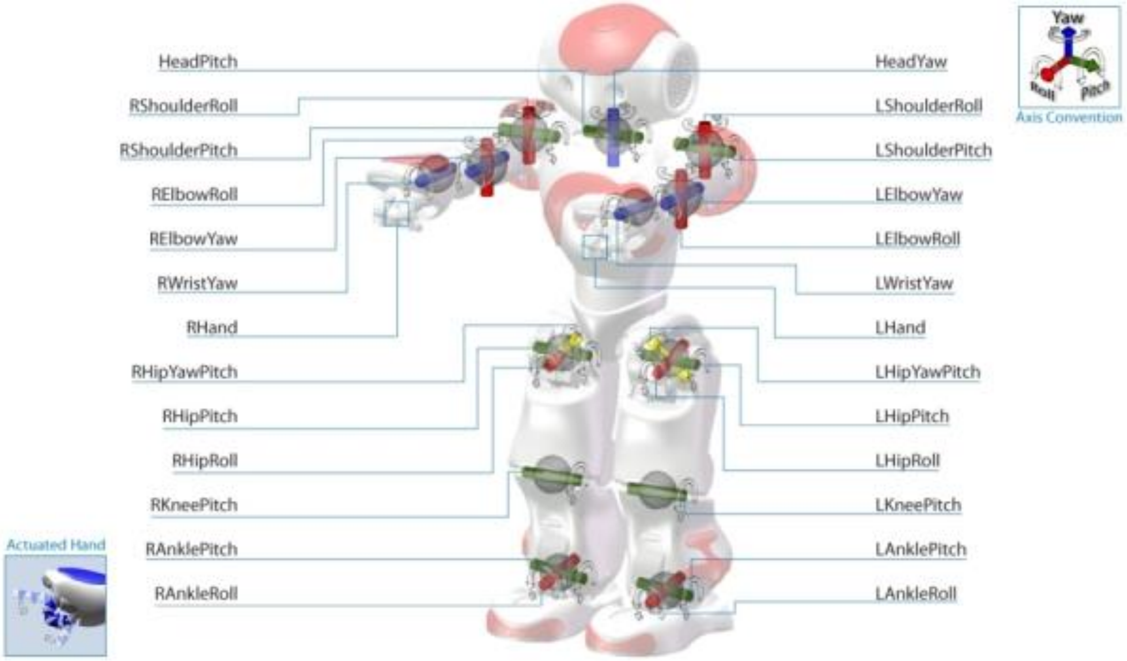


Figure 7. NAO H25 Joints [8].

Table 2. NAO's kinematic chains.

Name	Chain
Head	HeadYaw, HeadPitch;
Left Arm	LShoulderPitch, LShoulderRoll, LElbowYaw, LElbowRoll, LWristYaw, LHand.
Right Arm	RShoulderPitch, RShoulderRoll, RElbowYaw, RElbowRoll, RWristYaw, RHand.
Left Leg	LHipYawPitch, LHipRoll, LHipPitch, LKneePitch, LAnklePitch, LAnkleRoll.
Right Leg	RHipYawPitch, RHipRoll, RHipPitch, RKneePitch, RAnklePitch, RAnkleRoll,

The robot’s joints have a limited angle of movement (Table 3); and for the legs the movement was considered in such a way that they do not collide with each other [8].

Table 3. NAO’s joint information.

Name	**Length (mm)**	**Range (Deg)**	**Range (raD)**
HeadYaw	Head joint twist (Z)	-119.5 to 119.5	-2.0857 to 2.0857
RShoulderRoll	Right shoulder joint right and left (Z)	-94.5 to -0.5	-1.6494 to -0.0087
RWristYaw	Right wrist joint (X)	-104.5 to 104.5	-1.8238 to 1.8238
RShoulderPitch	Right shoulder joint front and back (Y)	-119.5 to 119.5	-2.0857 to 2.0857
RElbowRoll	Right elbow joint (Z)	0.5 to 89.5	0.0087 to 1.5621
HeadPitch	Head joint front and back (Y)	-38.5 to 29.5	-0.6720 to 0.5149
RElbowYaw	Right shoulder joint twist (X)	-119.5 to 119.5	-2.0857 to 2.0857
RHand	Right hand	Open And Close	
LHipRoll	Left hip joint right and left (X)	-21.74 to 45.29	-0.3794 to 0.7904
LWristYaw	Left wrist joint (X)	-104.5 to 104.5	-1.8238 to 1.8238
LAnkleRoll	Left ankle joint right and left (X)	-44.06 to 22.79	-0.7690 to 0.39780
LElbowRoll	Left elbow joint (Z)	-89.5 to -0.5	-1.5621 to -0.0087
LAnklePitch	Left ankle joint front and back (Y)	-68.15 to 52.86	-1.1895 to 0.9227
RHipYawPitch	Right hip joint twist (Y-Z 45°)	-65.62 to 42.44	-1.1453 to 0.74080
RKneePitch	Right knee joint (Y)	-5.90 to 121.47	-0.1030 to 2.1201
RAnkleRoll	Right ankle joint right and left (X)	-22.27 to 45.03	-0.3886 to 0.7858
LElbowYaw	Left shoulder joint twist (X)	-119.5 to 119.5	-2.0857 to 2.0857
LKneePitch	Left knee joint (Y)	-5.29 to 121.04	-0.0923 to 2.1125
LHipYawPitch	Left hip joint twist (Y-Z 45°)	-65.62 to 42.44	-1.1453 to 0.7408
RHipPitch	Right hip joint front and back (Y)	-101.54 to 27.82	-1.7723 to 0.4856
RKneePitch	Right ankle joint front and back (Y)	-67.97 to 53.40	-1.1864 to 0.9320
LShoulderRoll	Left shoulder joint right and left (Z)	0.5 to 94.5	0.0087 to 1.6494
LHipPitch	Left hip joint front and back (Y)	-101.63 to 27.73	-1.7739 to 0.4840
LHand	Left hand	Open And Close	
RHipRoll	Right hip joint right and left (X)	-42.30 to 23.76	-0.7383 to 0.4147
LShoulderPitch	Left shoulder joint front and back (Y)	-119.5 to 119.5	-2.0857 to 2.0857

Link Specification.

NAO Robot is 57 cm tall with an arm length of 29 cm. The measurement between the shoulders is 27 cm (Figure 8, box 1). The **X** axis is positive toward NAO's front, the **Y** from right to left and the **Z** is vertical (Figure 8, box 2). Links are defined using joint names and a point called Torso, located at 12,65 Z (cm) from HeadYaw.

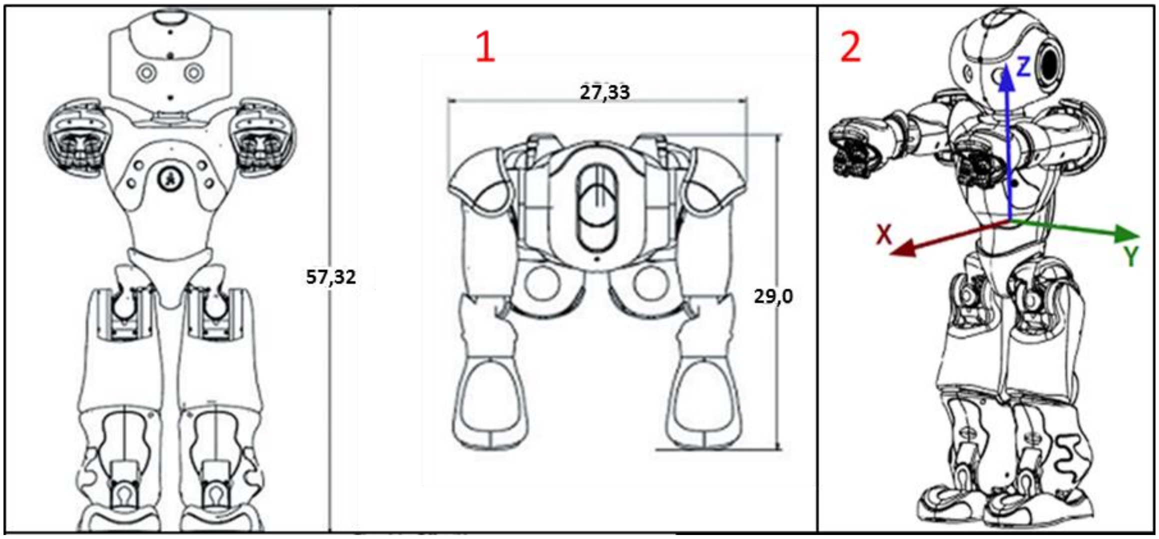


Figure 8. NAO's frame (1) Body lengths, (2) Axis definition [8]

NAO Robot is 126.5 mm from the its center of gravity to the its neck (Figure 9), 100 mm to the shoulder joint, 85 mm to the hip, and 100 mm to the thigh. It is 102.75 mm from the knee to the ankle, the foot height is 45.11 mm, the length between the neck and shoulder joint is 98 mm, and the hand width is 15.9mm (Table 4).

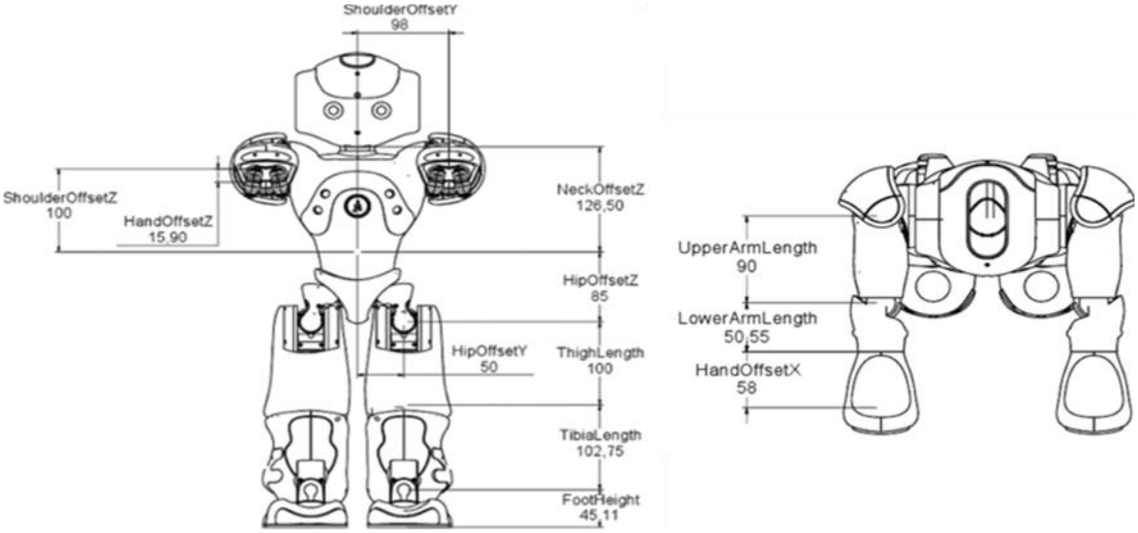


Figure 9. Lengths of NAO's Parts [8]

Table 4. Lengths of NAO's links.

Name	***Length (mm)***
NeckOffsetZ	126.5
ThighLength	100
ShoulderOffsetZ	100
UpperArmLength	90
FootHeight	45.11
HipOffsetZ	85
ShoulderOffsetY	98
TibiaLength	102.74
HandOffsetX	58
owerArmLength	50.55
HandOffsetZ	15.9
HipOffsetY	50

Vision System

There are two frontal cameras attached to NAO head (Figure 10, box 1); the top camera focuses on the front whereas the bottom camera focuses on the foot (Figure 10, box 2). They provide a 640x480 resolution at 30 frames per second (Table 5).

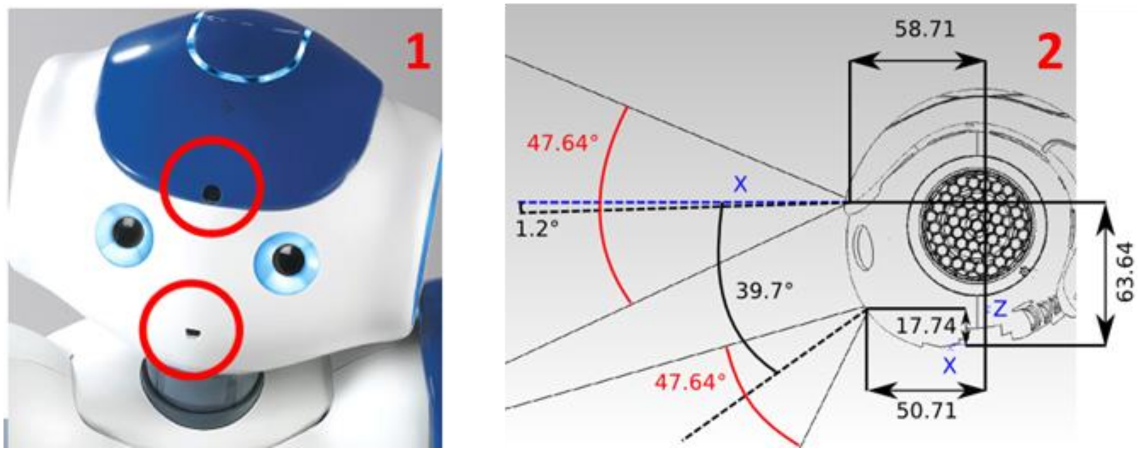


Figure 10. NAO vision system: (1) cameras placemen, (2)location and angle representation[8].

The vision system can be used to implement mark recognition, face recognition, object recognition, image recording, etc. Traditional stereo vision cannot be constructed directly using the cameras on the NAO because they have no overlapped field of view [48].

Table 5. NAO vision system frame rates.

Name	**Local**	**Gb Eth**	**100Mb Eth**	**WiFi g**
160x120 (QQVGA)	30fps	30fps	30fps	30fps
320x240 (QVGA)	30fps	30fps	30fps	11fps
640x480 (VGA)	30fps	30fps	12fps	2.5fps
1280x960 (4VGA)	10fps	10fps	3fps	0.5fps

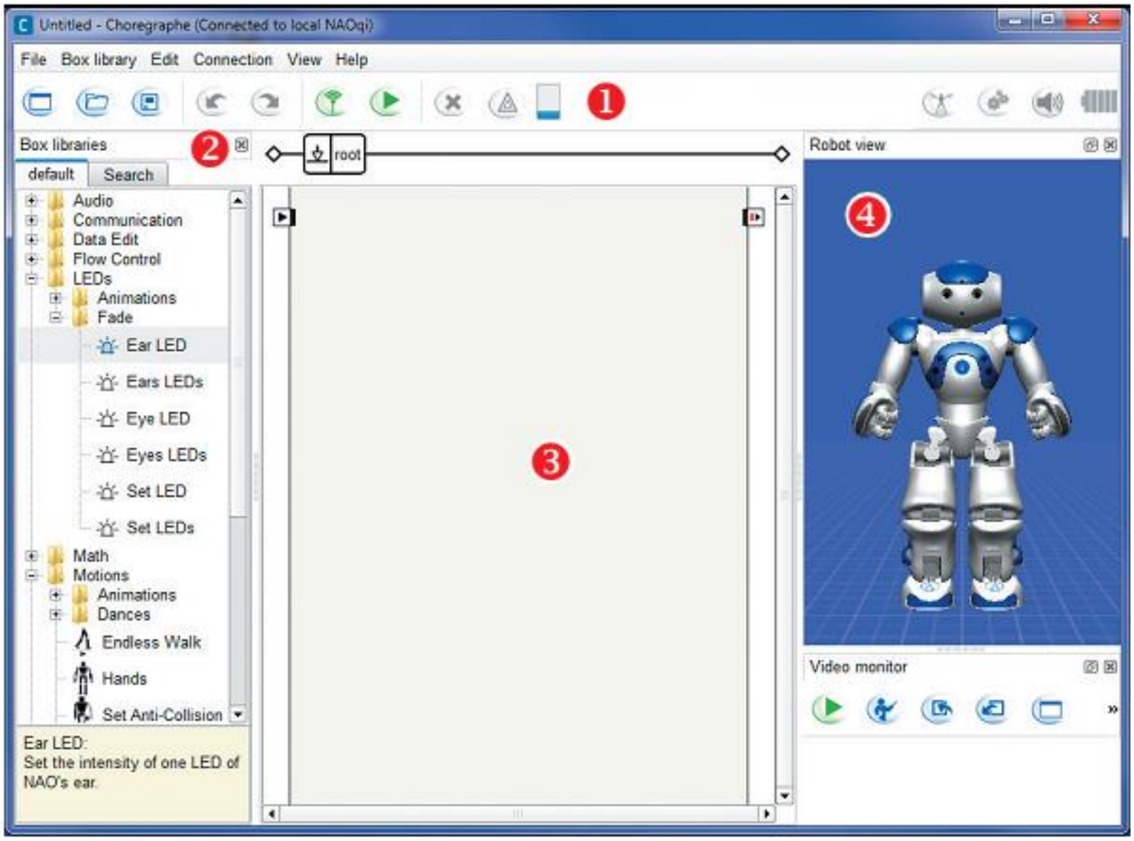
NAO's Software.

Nowadays NAO Robot's users are mainly research laboratories; however, Aldebaran Robotics wants to spread its product to a more general audience [49]. For instance, teachers and therapists need to be enabled to control the robot using a friendly and known method; this means less time for technicalities and more time dedicated to the students or patients [37]. It was necessary to supply to the NAO Robot with a powerful programming environment [49]; therefore it is provided with **Choregraphe**, **NAOqi**, and **Monitor** as development software.

Choregraphe is a graphics-based software tool, it is easy to use even by beginner users [8]; it allows to create complex applications and fine control of motions [49]. **NAOqi** is the programming framework used that manages the modules execution. These modules consist of programming libraries that can be created and executed within the robot [50]. On the other hand, **Monitor** is a program that receives feedback from the NAO Robot allowing easily to verify joint or sensor values[8].

Choregraphe

Choregraphe is a very simple and intuitive graphical environment that allows programming the NAO Robot [49]. This is an all in one humanoid robot features embedded Software. Unlike text-based programming, the graphics-based programming has a lower focus on grammar, and programming is mostly achieved by using a mouse rather than a keyboard to create codes [8]. The versatility of its agnostically platform (designed to be compatible across most common systems) allows the control of programs with Windows, MAC OS and even Linux [37]. When installing Choregraphe in the Windows operating system, both NAOqi and Monitor are installed together.



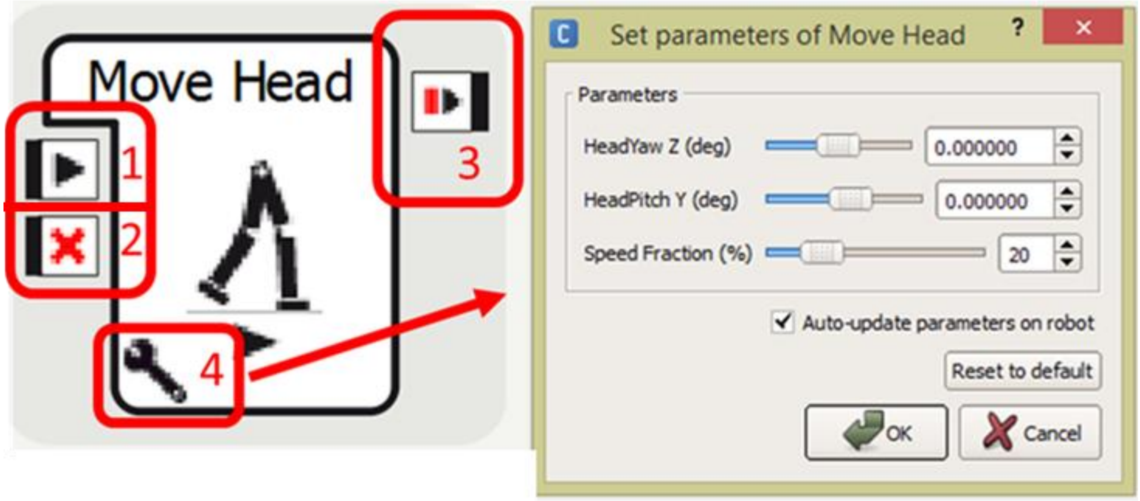
Ear LED:
Set the intensity of one LED of NAO's ear.

Figure 11. The Choregraphe Graphic Interface. (1) Menu, (2) Box Library, (3) Flow diagram Space and (4) 3D graphical representation of NAO.

The main interface is displayed at the startup (Figure 11). The interface is divided into four zones: Zone 1, **Menu**, there is a drop-down menu with File, Edit, Connection, Behaviors, View, and Help, and so on (Figure 11, box 1). Zone 2, the **Box Library** that contains the list of available behaviors (Figure 11, box 2). Here, a box refers to an icon with a function. The box library is largely divided into 13 types depending on function and is made up of 70 or so boxes [8]. Zone 3, the **Flow diagram Space,** where NAO's movements are created (Figure 11, box 3), the user can drag-and-drop the boxes in the box library here. It allows the user to graphically place behaviors composed of library boxes and links between them [49]. In Zone 4, a **3D graphical** representation of NAO is able to execute the implemented behavior (Figure 11, box 4). It simulates the movement programmed by the user. This function is used only for joint movements, and simulation, it does not work to use other robot's elements such as sensors, LEDs, and cameras.

A behavior is a piece of software that controls the robot. Programming NAO consists in implementing new behaviors through the combination of several behaviors [49]. Boxes and icons are an important element of Choregraphe and are placed in the diagram. Graphics-based programming is done through connecting these boxes [8] in series or parallel through their inputs and outputs [49].

The box is configured with an input, output, and the Parameter button (Figure 12). Some boxes don't have an output or the Parameter button [8]. To execute a block, a signal must be present on the "**OnStar"** input (Figure 12, box 1), at the end of execution; a signal could be present in the "**onStopped"** output (Figure 12, box 3). This signal can easily become the input for another block. A signal on "**onStop"** input (Figure 12, box 2) stops the box functions. The "**parameter**" button (Figure 12, box 4), allows to change or create some settings about the action made by the box [8].



Set parameters of Move Head

Parameters

HeadYaw Z (deg) 0.000000

HeadPitch Y (deg) 0.000000

Speed Fraction (%) 20

☑ Auto-update parameters on robot

Reset to default

Figure 12. Choregraphe Box. (1) onStart input, (2) onStop input, (3) onStopped output and (4) Parameter button.

If a real NAO (physical NAO robot) is connected (via Wi-Fi or Ethernet) to the PC running Choregraphe, the real NAO will execute the global behavior. When no real robot is available, the virtual NAO (or simulated NAO) can execute the programmed behavior [49].

NAOqi

The efficiency of Choregraphe is due to the NAOqi which is a running framework behind Choregraphe [49]. It was developed specifically by Aldebaran Robotics to use the NAO Robot, this includes: parallel processing, resources management, synchronization, and event processing, element which are generally required for robotics [8].

These modules are programming libraries that can be created and executed on the robot. Some modules are available with the robot to control motions, LEDs, text-to-speech, etc. [50], nevertheless, is possible to create customized modules.

NAOqi is cross-language, it allows Python , Urbi or C++ interpreters to use NAOqi’s features through a dedicated SDK. The NAOqi architecture is event-driven, a lot of effort was required to make the architecture support both parallel and sequential calling methods [51].

Chapter 4

NAO Robot's Interactive Applications

Robots have been used on many fields as programmable machines for execution of preprogrammed tasks such as painting, welding, simple assembly task, etc. [7], [52]. In different fields of research, there have been designed many systems to interact with people; these systems can embody a robot (or an animated avatar) or not (the human interacts through speech or text entered at a keyboard) [52]. In recent years, robots with different appearances and functions have been used in education, medical diagnosis and treatments [53].

The objective of this chapter is to develop a series of interactive applications to grant the NAO Robot the capabilities of: Teleoperation, facial and hand gestures recognition and expression of emotions. All of these applications were developed in Python and tested independently.

NAO Robot Teleoperation

Progress in robotic and telecommunications technologies has been significant lately [11]. Autonomous robots are generally designed to function remotely as possible and without the direct intervention of the user; often performing tasks in dangerous and hostile environments [52]. Usually, a wireless internet platform is used to control the robot. Internet platforms support the connection to the IP network via the mobile communication network (smart phones - tablets) or WIFI [54].

Although in robotics, automation would be ideal, teleoperation is essential especially when robot is operating in a changing scenario. However, teleoperation has some problems due to time delay and sensory perception between the operator and the robot [55]. The operator cannot recognize the state of the robot and environment at the same moment, hence, sensors use becomes fundamental.

To control and/or to Access the robot remotely, it was developed a tele-control system based on WWW technology. It communicates with a remote client via the client-server internet type using the Internet communication protocol TCP / IP. Components of this control system (Figure 13) include the **HTTP Server** on port 800, the **Web Socket Server** on port 8000 and **Video Server** on port 5003

The Client Host can access to the control interface through the HTTP server. Real time video (frames from cameras) can be reachable into control interface or via Video Server directly. The Web Socket Server exchanges information with embedded socket client, this create a tunnel between the client host and the NAO Robot allowing to send control commands. Ports 22, 23 and 80 are reserved for services ssh, telnet and web page of NAO Robot respectively. A Python script controls each module.

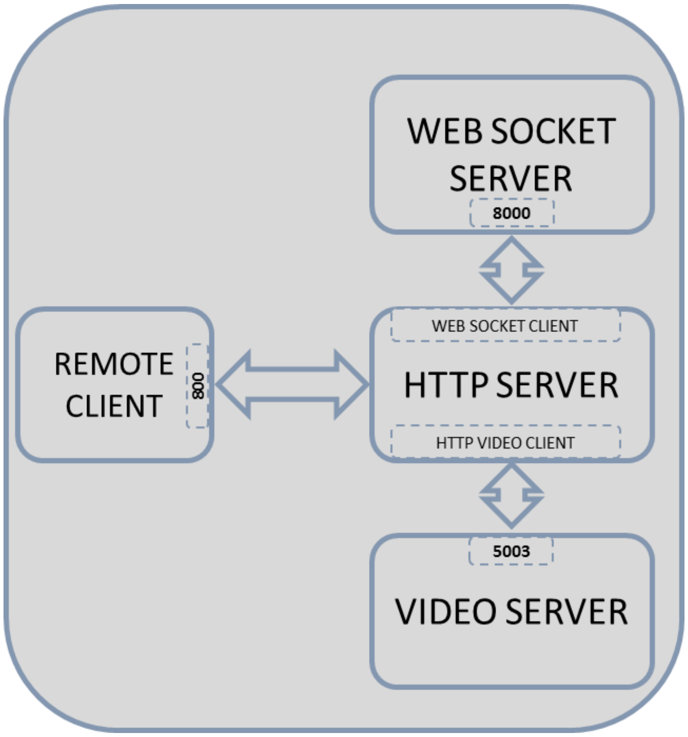


Figure 13. Detailed structure of the teleoperation system.

The communication can be established in three possible scenarios (Figure 14). The internal or **LAN** (Figure 14, box 1), in which the robot and the host are within the same network domain; the external or **WAN-Internet** in which the host and the robot are in different network domains, external communication may be with a **Public IP** address (Figure 14,box 2) or **Private IP** address (Figure 14, box 3).

In scenarios 1 and 2 (Figure 14, box 1 & 2), the connection is established directly via the IP address and port. In the first case, the user can only access if the host is in the same domain (subnet) of the robot, whereas in the second one, the user can access from any host with Internet access. However, in the case of the scenario 3 (Figure 14, box 3), it is required to port forwarding the edge router to successfully establish the communication.

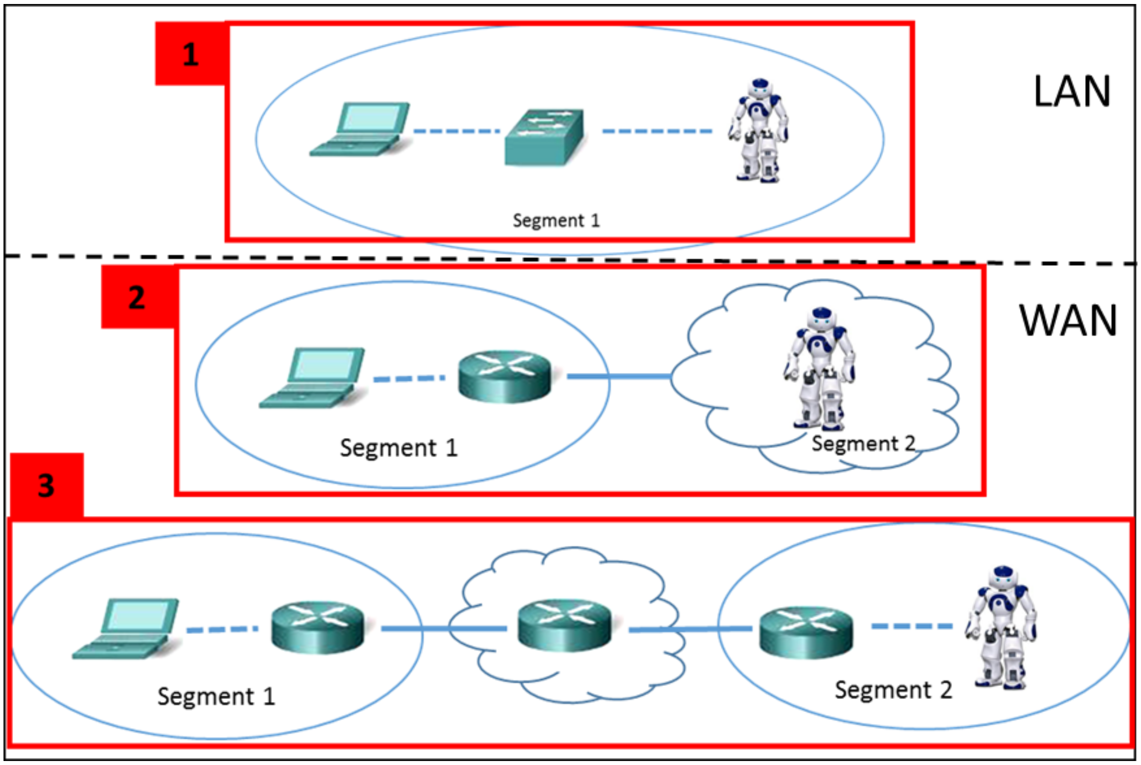


Figure 14. Tele-operation system architecture. (1) LAN, (2) WAN - Public IP and (3) WAN - Private IP. The cloud indicates a connection through the Internet.

A client host can access web control interface (Figure 15) through one of the many web browsers (e.g. Google Chrome) by entering the IP address assigned to the robot and the port number 800 instead of Internet address (e.g. 192.168.1.21:800). In this case, the port number will always be 800; however, the IP address can be configured on the robot manually or via DHCP server. If using DHCP, the user may want to use a fixed DHCP assignment, thus, the server always will deliver the same IP address to the NAO Robot.

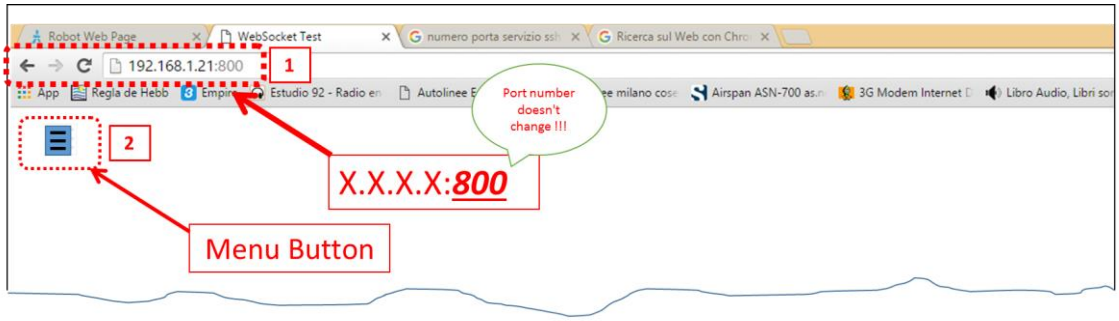

Robot Web Page
WebSocket Test
numero porta servizio ssh
Ricerca sul Web con Chrome
192.168.1.21:800
1
2
Port number doesn't change !!!
X.X.X.X:800
Menu Button

Figure 15. Control interface access. (1) NAO Robot's IP address and port and (2) Web control interface Menu Button.

By clicking on the menu button, the connection could be established by using the Connect button (Figure 16, box 1); then, all the commands to control the NAO Robot will be enabled. The connection can be closed using the Disconnect button (Figure 16, box 2).

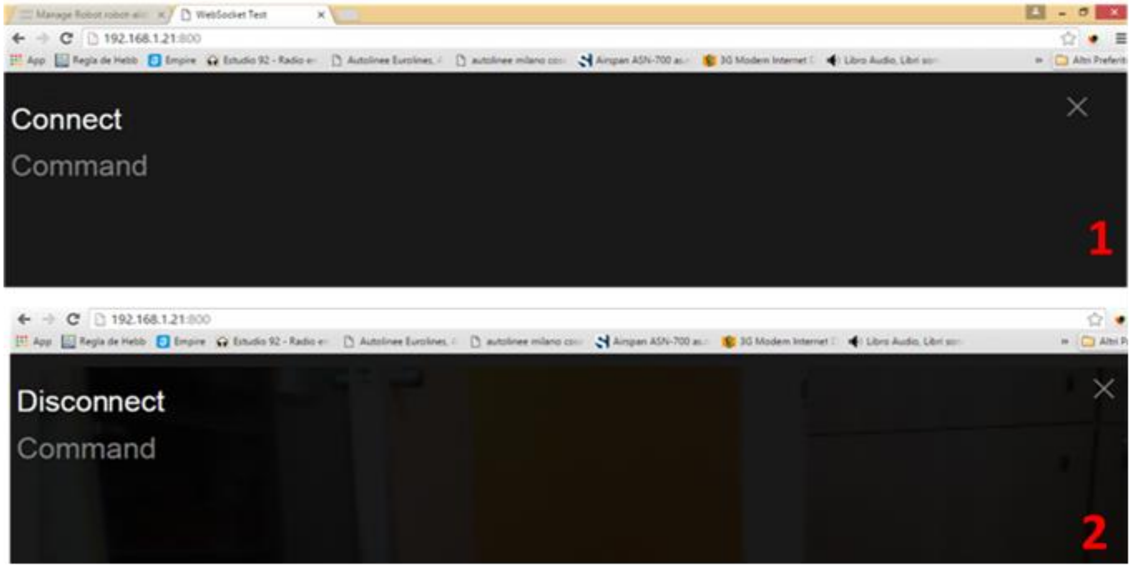


Figure 16. Menu window. (1) Connect button option and (2) Disconnect button option.

The Command button displays a list of all the commands available (Figure 17), the command is sent when clicking on.

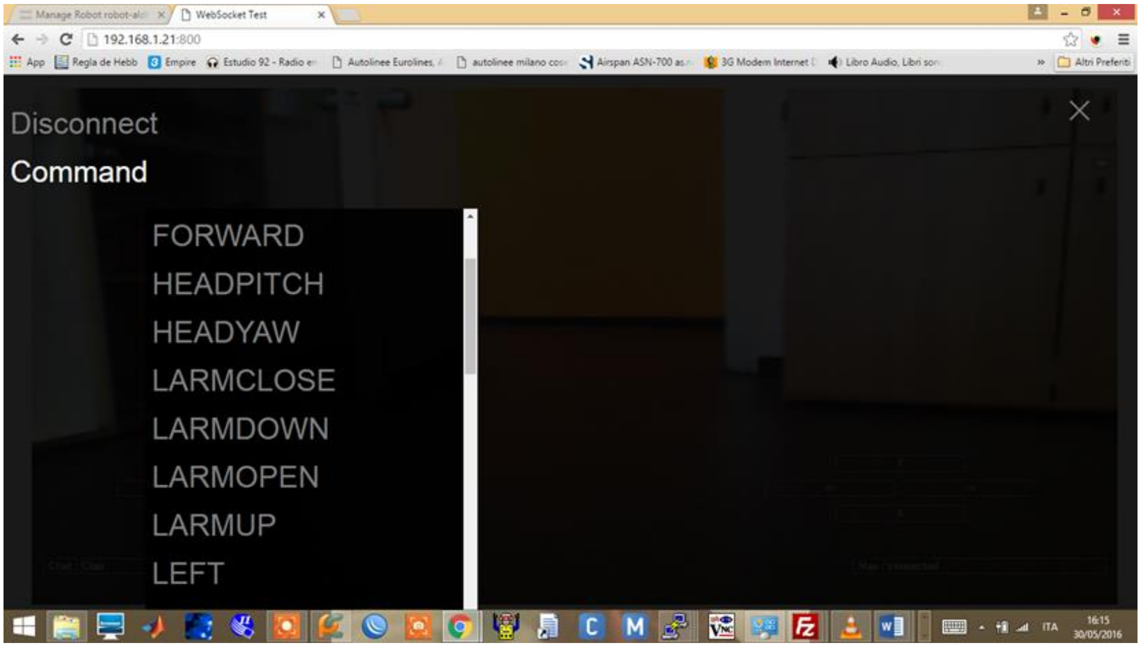
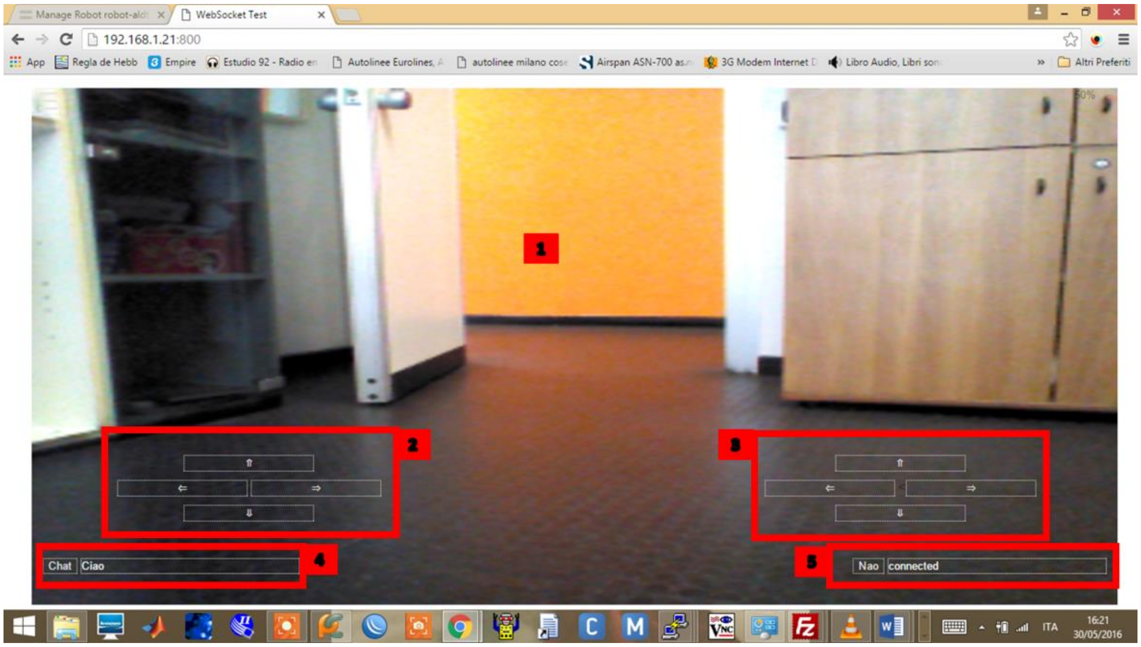


Figure 17. Commands list.

Close the menu to access to the video mode, a new main window appears (Figure 18). The background shows real-time images from NAO Robot’s camera (Figure 18, box 1). Movement commands (Figure 18, box 2) allow forward-backward displacement and left-right turns, head control commands (Figure 18, box 3) allow to move the head horizontally and vertically. The Chat (Figure 18, box 4) allows to send a text and synthesizes it through the audio output of the robot. Section Nao (Figure 18, box 5), allows visualizing messages to and from the robot to check its correct operation.



Chat Ciao

Nao connected

Figure 18. Control interface Main window. (1)Real Time images frames, (2) Body motion control, (3), Head motion control, (4) Chat box, (5) Messages Box (video link 1- 2).

The most common scenario is the WAN with Private IP. In this case, there is an edge router, which is visible to outside (Figure 19). Commonly, SOHO (Small Office / Home Office) routers are used for standard home networking (networks that support between 1 and 10 people). To configure the router port forwarding, the steps are the same; however, the Web user interface for managing could be different and depends on the router model. In this case, a NetGear WNR2000 is used. To access to the router Web interface, the router IP address (default IP if no changed) and security credentials (default if no changed) are needed.

The first step is the configuration of the static DHCP, it allows to assign a unique fixed IP address to a specific device on the network (Figure 20); then the port forwarding can be set up.

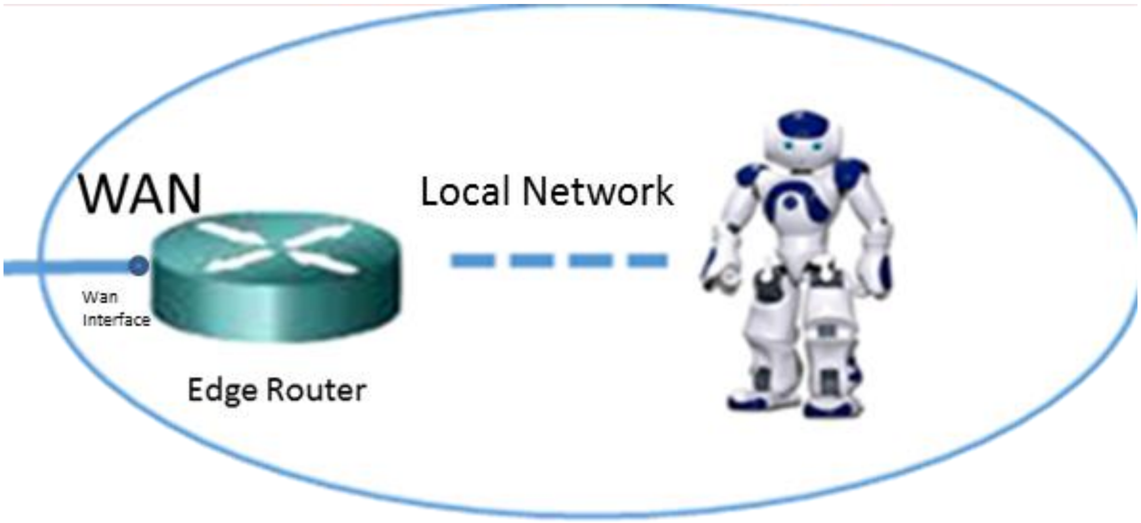
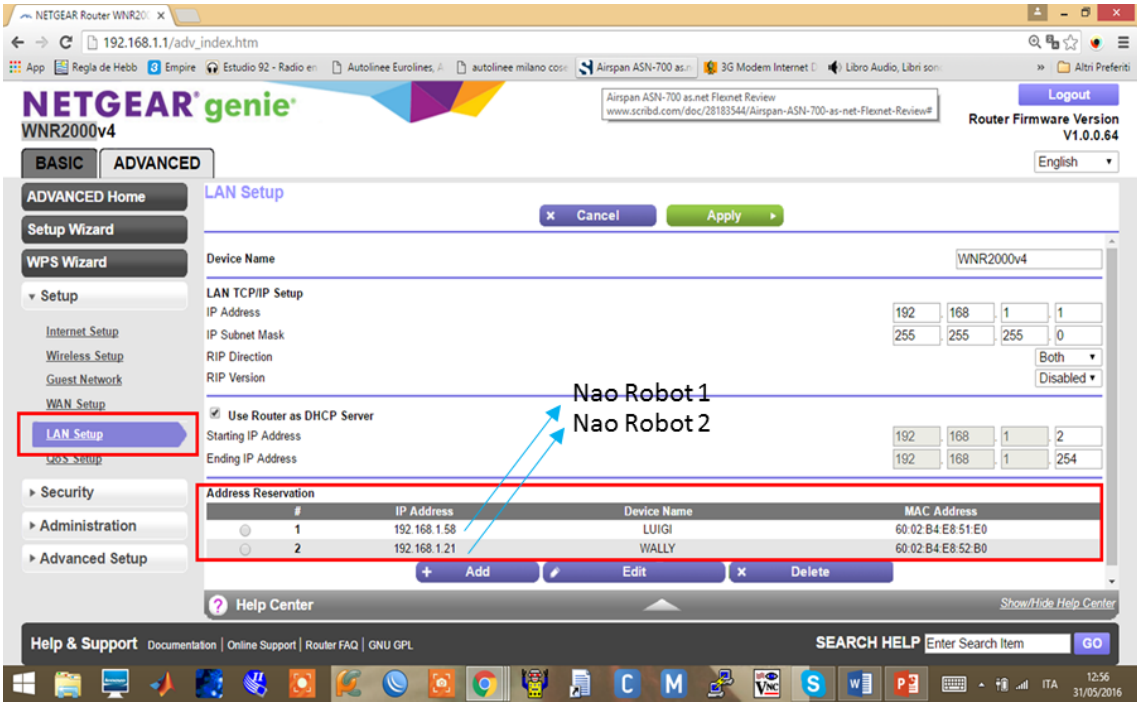


Figure 19. WAN access scenario. Remote site should be able to reach the Wan interface of the Edge router.



LAN Setup

Device Name: WNR2000v4

LAN TCP/IP Setup

IP Address: 192 . 168 . 1 . 1

IP Subnet Mask: 255 . 255 . 255 . 0

RIP Direction: Both

RIP Version: Disabled

☑ Use Router as DHCP Server

Starting IP Address: 192 . 168 . 1 . 2

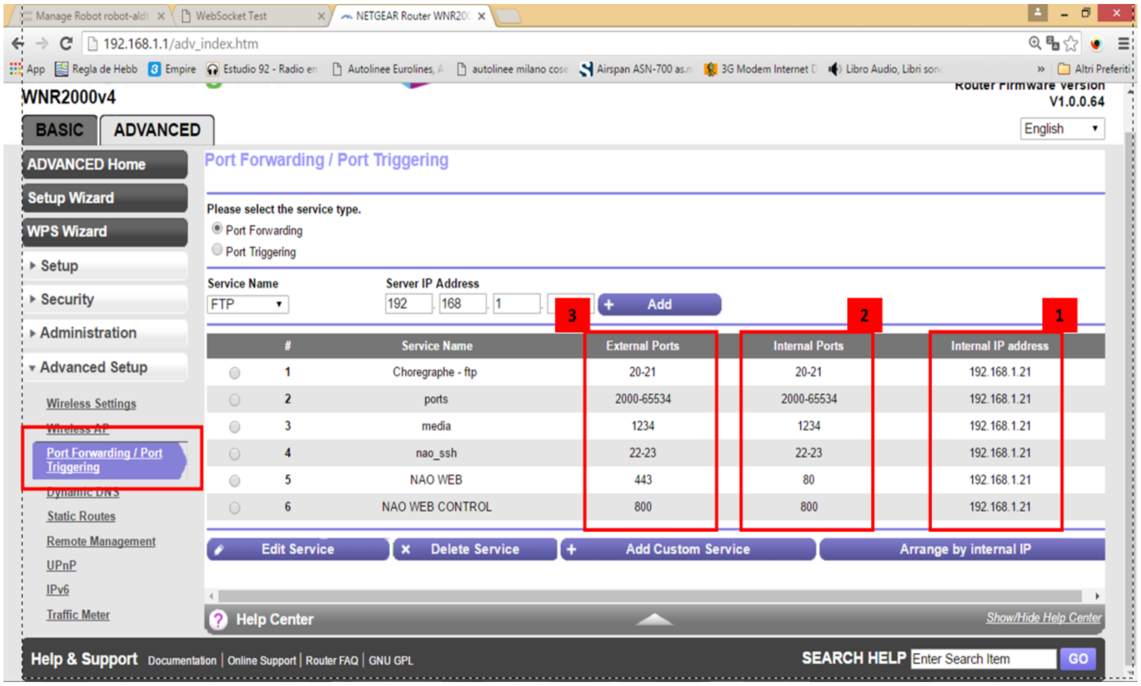
Ending IP Address: 192 . 168 . 1 . 254

Address Reservation

	#	IP Address	Device Name	MAC Address
○	1	192.168.1.58	LUIGI	60:02:B4:E8:51:E0
○	2	192.168.1.21	WALLY	60:02:B4:E8:52:B0

Figure 20. Static DHCP configuration to assign the same IP address to NAO Robot.

To set up the Port forwarding, some elements need to be established: the external ports on which the router has to listen to connections (these ports must to be open), the internal ports used by the tele-control system and the IP address assigned to the robot from the DHCP server (Figure 21).



192.168.1.1/adv_index.htm

WNR2000v4

Router Firmware Version V1.0.0.64

Port Forwarding / Port Triggering

Please select the service type.

● Port Forwarding
○ Port Triggering

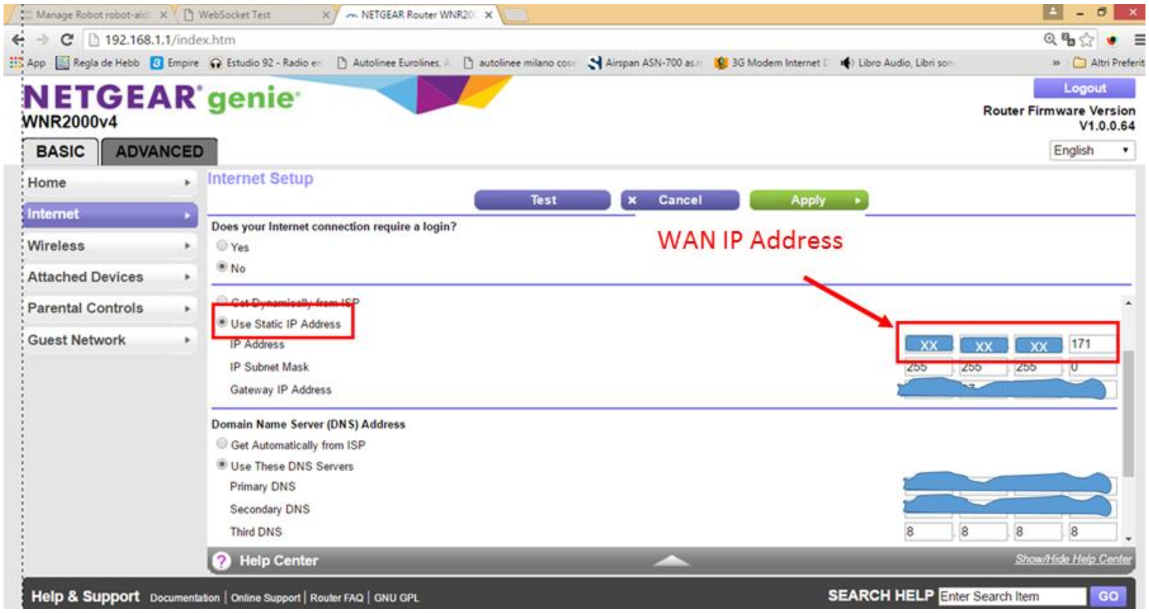
Service Name: FTP

Server IP Address: 192 . 168 . 1 .

	#	Service Name	External Ports	Internal Ports	Internal IP address
○	1	Choregraphe - ftp	20-21	20-21	192.168.1.21
○	2	ports	2000-65534	2000-65534	192.168.1.21
○	3	media	1234	1234	192.168.1.21
○	4	nao_ssh	22-23	22-23	192.168.1.21
○	5	NAO WEB	443	80	192.168.1.21
○	6	NAO WEB CONTROL	800	800	192.168.1.21

Figure 21. Port forwarding configuration. (1) External Ports, (2) Internal Ports and (3) Robot IP Address.

In the Internal IP address field, the assigned robot's IP address is entered (Figure 21, box 1). The internal TCP ports (Figure 21, box 2) used are: HTTP Server: 800, SSH and Telnet: 22-23, NAO Robot's Web page: 80, FTP service: 20-21. Other ports 5003 and 8000 used by the video socket server and Web server socket, respectively, these are in range of enabled ports in Ports section: 2000-65534. For external ports (Figure 21, box 3) can be used any free port number. However, it is possible to use the same internal port number, as long as it is available. Finally, checking the WAN IP address of the edge router which can be reached from the outside (Figure 22). If the WAN IP address is dynamically assigned, it should be controlled periodically.



Internet Setup

Does your Internet connection require a login?

○ Yes
● No

○ ~~Get Dynamically from ISP~~
● Use Static IP Address

IP Address XX . XX . XX . 171
IP Subnet Mask 255 . 255 . 255 . 0
Gateway IP Address [redacted]

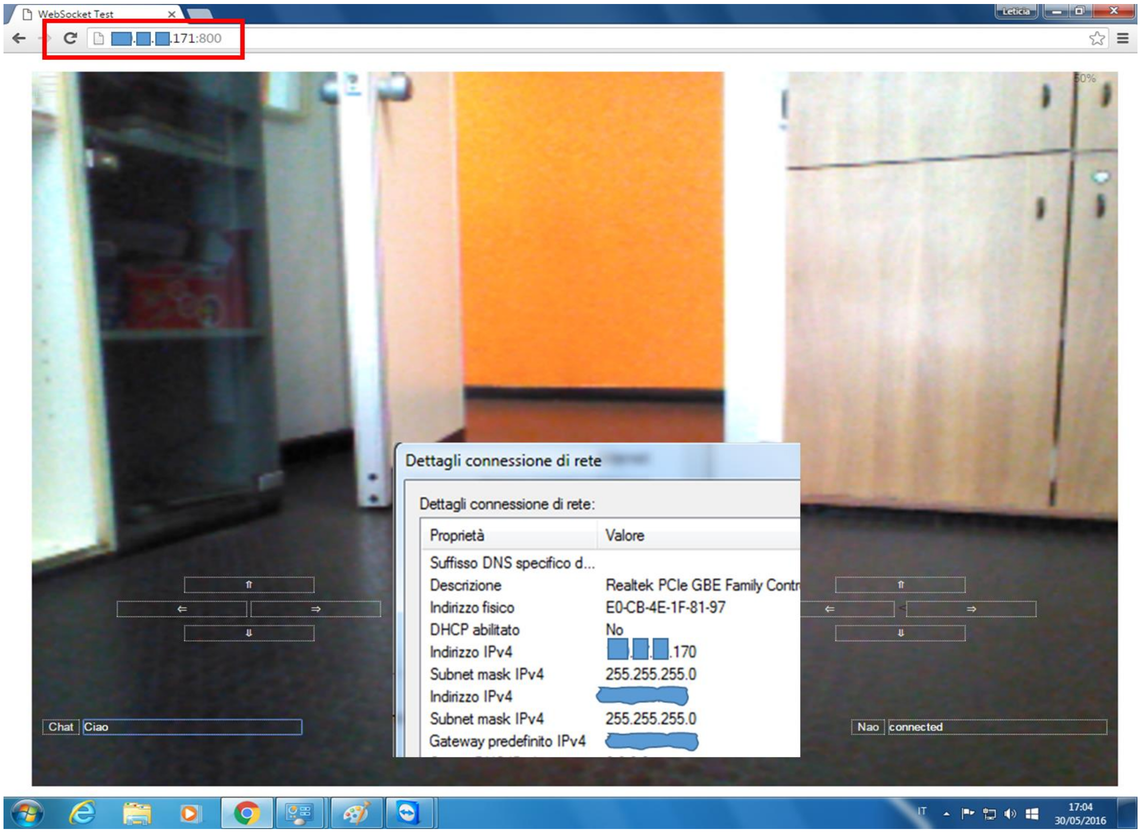
Domain Name Server (DNS) Address

○ Get Automatically from ISP
● Use These DNS Servers

Primary DNS [redacted]
Secondary DNS [redacted]
Third DNS 8 . 8 . 8 . 8

Figure 22. Router WAN IP status.

At this point, the NAO Robot can be acceded from any remote point that is able to reach the WAN interface of the edge router (Figure 19) ; either by routing or via Internet (public IP address in the router) (Figure 23) .



[redacted].[redacted].[redacted].171:800

Chat: Ciao

Nao: connected

Dettagli connessione di rete

Dettagli connessione di rete:

Proprietà	Valore
Suffisso DNS specifico d...	
Descrizione	Realtek PCIe GBE Family Contr
Indirizzo fisico	E0-CB-4E-1F-81-97
DHCP abilitato	No
Indirizzo IPv4	[redacted].[redacted].[redacted].170
Subnet mask IPv4	255.255.255.0
Indirizzo IPv4	[redacted]
Subnet mask IPv4	255.255.255.0
Gateway predefinito IPv4	[redacted]

Figure 23. .Access to the NAO Robot from outside to the robot's local network.

Face recognition.

In **Human-Robot Interaction** (HRI), face detection provides very significant information for the interaction. Face detection is able to detect people faces through tracking the image by various methods [56]. On the other hand, face recognition can be done by scanning a face and comparing the facial features with the imagines from a database [57].

The NAO Robot has a vision API (application programming interface) **ALFaceDetection** which allows to the NAO Robot to detect, and optionally recognize faces [8]. The API analyzes the basic features and provides information about position and shape of the face. It also generates 31 landmarks (Table 6) for a face representation including the contour of the mouth, nose position, shape of each eyebrow and contour for each eye [58]. If the information provided by NAO Robot's vision APIs is not enough, it could be used remotely in order to increase its performance by using an external tools software.

Table 6. ALFaceDetection landmarks.

Name	**#points**
Mouth	8
Nose	3
Eyebrow	3/3
Eye	7/7

On the other hand, Python OpenCV [59] contains many pre-trained classifiers as XML files for facial detection (Figure 24), eyes, smile etc.; however, it is possible to train a new classifier for any object such as cars, planes etc. [59]. It is a "Haar feature-based cascade" type classifier witch is an effective object detection method [60]. In this approach, a machine learning is trained using a set of positive and negative images [59]. Features, are extracted from the image data and the level of similarity between two images can be evaluated based on distances between the images' corresponding features [61].

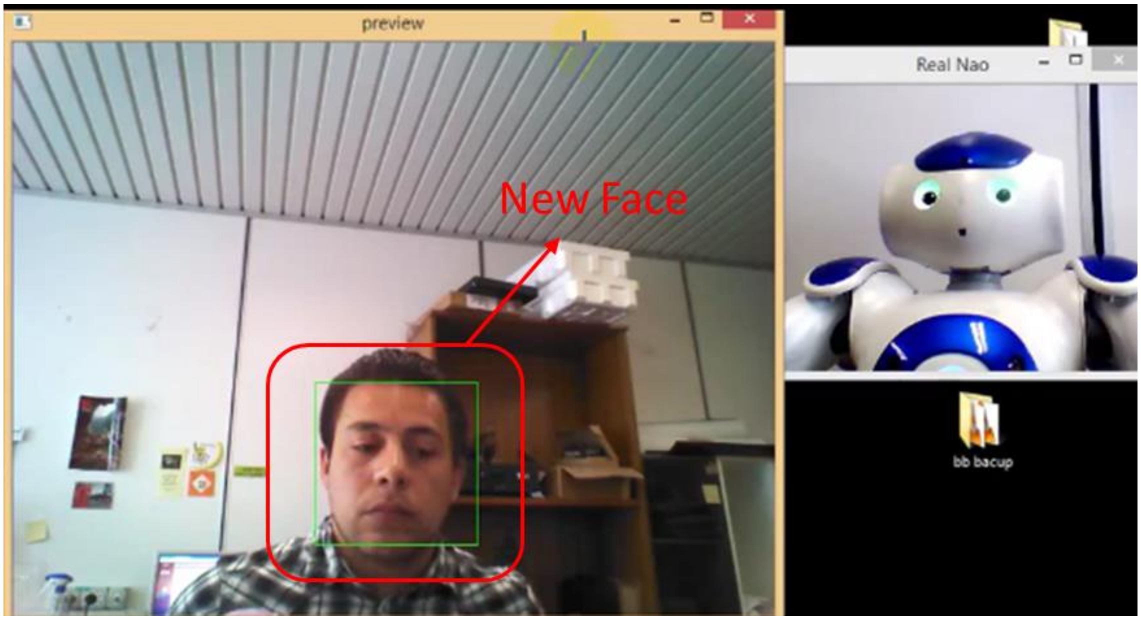
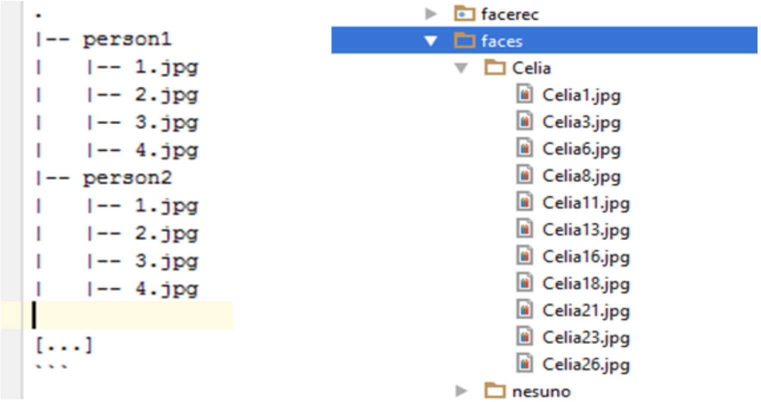


Figure 24. Face detection test using real NAO and Open CV *Haar feature-based cascade classifier* .

Additionally, NAO Robot has a Face Recognition module which provides to the robot the possibility not only to detect but also recognize people [8]. A learning stage is necessary to create the database. The database is stored in a specific directory on the robot; dedicated buttons in the video monitor allows to share, delete, export and import the database [8].

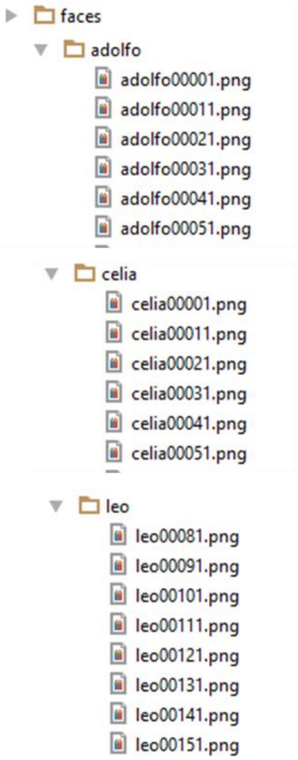
However, the NAO Robot's Recognition module is not very robust and the detection success depends on the angle, rotation and distance of the face; furthermore, the database could be modified only during runtime (only on the real robot). Due to these limitations, a different approach has been adopted in this work, where the database is composed by a set of images sorted by names corresponding to people that NAO Robot may recognize. The face images are stored in a hierarchical tree structure (Figure 25). Unlike the original NAO's database, using this approach make it possible to modify elements by simply adding images to the appropriate files (Figure 26); therefore, it is not necessary to do at runtime (the virtual robot can be used) and the images can be of different size.



```
.
|-- person1
|    |-- 1.jpg
|    |-- 2.jpg
|    |-- 3.jpg
|    |-- 4.jpg
|-- person2
|    |-- 1.jpg
|    |-- 2.jpg
|    |-- 3.jpg
|    |-- 4.jpg
|
[...]
```
```

```
▶ facerec
▼ faces
  ▼ Celia
      Celia1.jpg
      Celia3.jpg
      Celia6.jpg
      Celia8.jpg
      Celia11.jpg
      Celia13.jpg
      Celia16.jpg
      Celia18.jpg
      Celia21.jpg
      Celia23.jpg
      Celia26.jpg
  ▶ nesuno
```

Figure 25. DataBase structure of images stored in a hierarchical tree structure.



```
▶ faces
  ▼ adolfo
      adolfo00001.png
      adolfo00011.png
      adolfo00021.png
      adolfo00031.png
      adolfo00041.png
      adolfo00051.png
  ▼ celia
      celia00001.png
      celia00011.png
      celia00021.png
      celia00031.png
      celia00041.png
      celia00051.png
  ▼ leo
      leo00081.png
      leo00091.png
      leo00101.png
      leo00111.png
      leo00121.png
      leo00131.png
      leo00141.png
      leo00151.png
```

Figure 26. Face DataBase, of images of people that NAO may reconzine.
```

The standard image representation shows a problem because of its high dimensionality. For example a two-dimensional 100 × 100 grayscale image span a 10, 000 dimensional vector space, so it requires a significant increase of the computational resources which are usually limited in an embedded system. Alternatively, the dimensionality of the data can be reduced by converting a set of correlated variables into a set of smaller, uncorrelated variables, therefore all the information will be represented with a smaller number of useful dimensions [62]. This procedure is called **Principal Components Analysis** (PCA). The aim of PCA is to describe the maximum amount of variance with as few principal components as possible [63].

When a PCA is applied to face images, the eigenvalues and their corresponding eigenvectors are calculated [63]. These **eigenvectors** are designated as **eigenfaces** (Figure 27)**.** A distance measure for face recognition can be defined. This distance measure is defined between a template image in a training set and a given image in the test set [64].

The **Eigenfaces** method corresponds mainly to the PCA analysis with a Nearest Neighbor model. Eigenfaces encode facial features as well as illumination in images [62] therefore, this method retains unwanted variations due to lighting and facial expression.

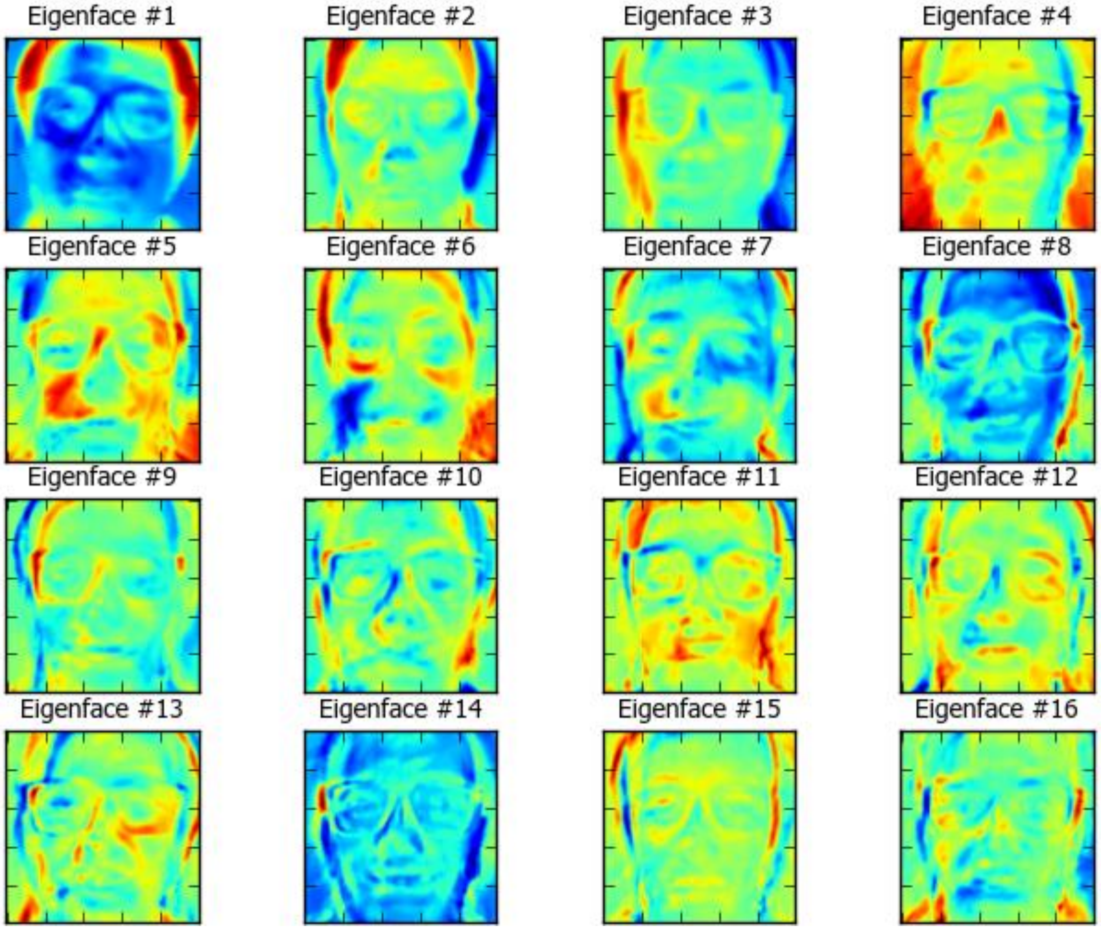


Figure 27. DataBase Eigenfaces, only 14 from 60 are showing Eigenfaces encode facial features and illumination.

Unlikely for face recognition, the **Discriminant Analysis** (DA) or **Fisherfaces** method is frequently used. In this case, a model is trained with a class-specific transformation matrix to classify between person by analyzing the facial features [65]. However, Fisherfaces method (Figure 28) critically depends on the quality of input data. Thus, if only well-illuminated pictures are using to learn faces, the recognition scene must be well illuminated too; otherwise, this method will probably fail [62]. Nevertheless, it is possible to improve total performance by using a PCA to reduce the dimension of the feature space and the standard DA for the classification task.

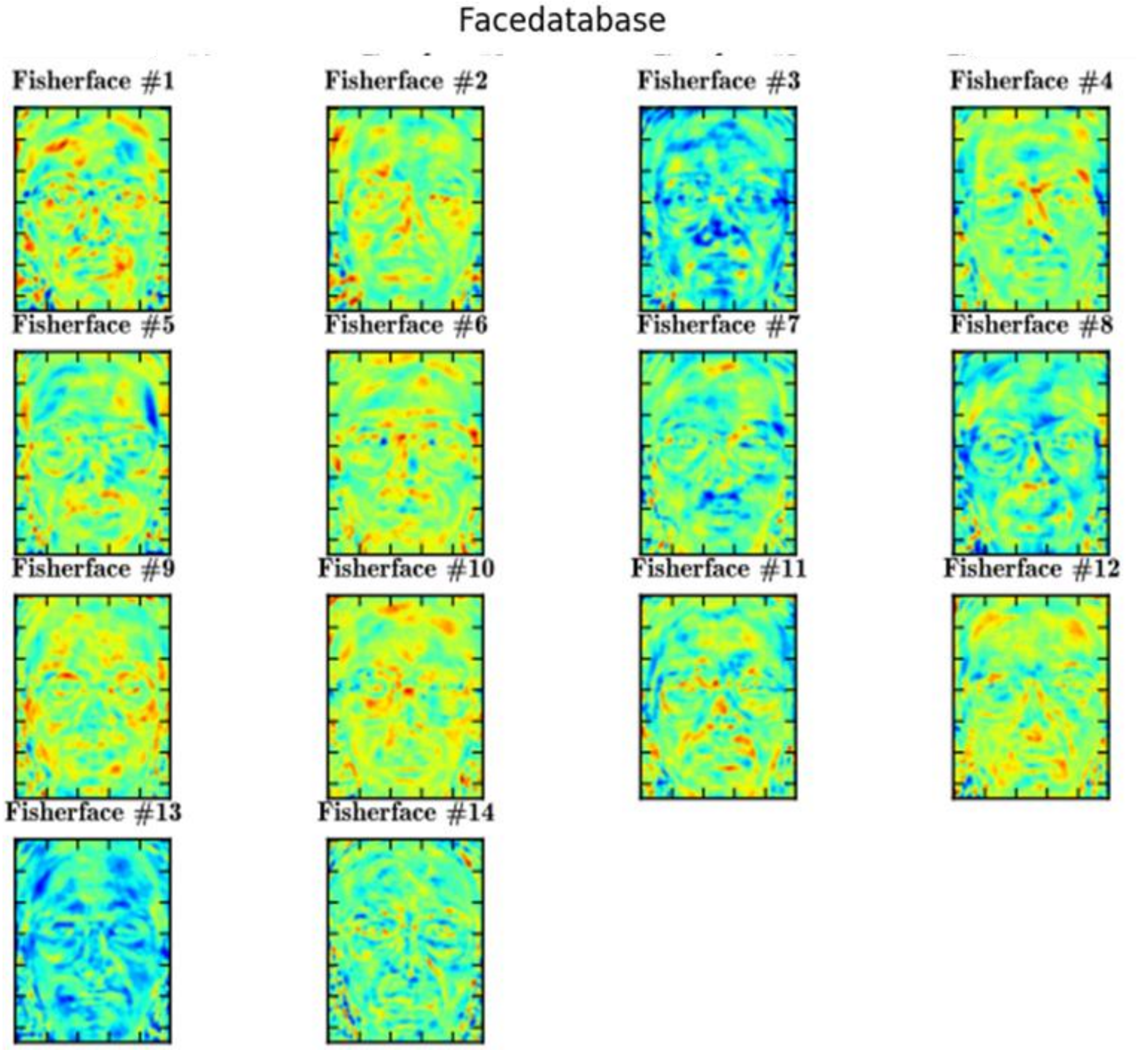


Figure 28. DataBase Fisherfaces, only 14 from 60 are showing. The performance of the Fisherfaces heavily depends on the input data as well.

A face recognizing machine (Figure 29) was developed using Python. The database was composed by 261 images of three people which should be identified by the NAO Robot. Images were taken by the NAO Robot under natural conditions; that is to say, the images were captured without the optimization of the illumination conditions (faces can show variation due to environment illumination changes, e.g. face alignment, facial expression, etc.).

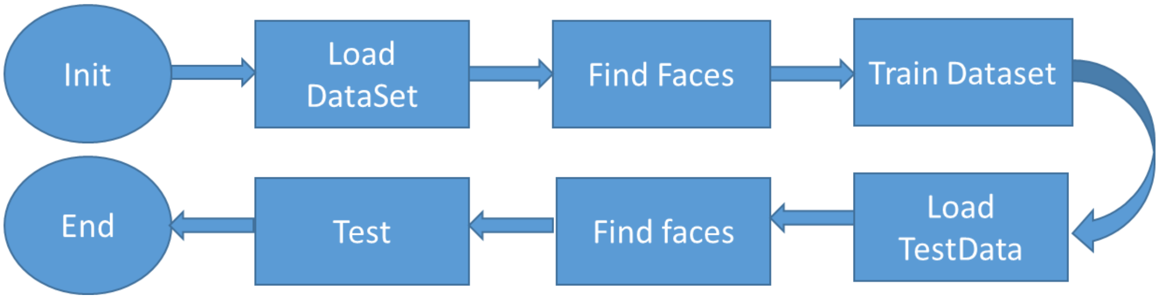


Figure 29. Classifier machine flow diagram. Before training and testing the model, the images in which the faces are not detected are discarded.

A simple reading function is used to load the images from all subfolders of a given database directory. Each directory is assigned a unique label (integer) that corresponds to the name of the folder that contains the images of each person. The function returns the resized images and the corresponding classes. Before model creation, the faces should be detected and extracted from the images, the cascade method mentioned before was used. After this step, the faces data set was reduced to 228 images (Figure 30). Once the data set is ready, it can be used to define the model, to learn the faces and to generate predictions. In order to test the model, a completely different data set was used. From 126 images; 85 faces were detected with a recognition success of 80% (Figure 31).

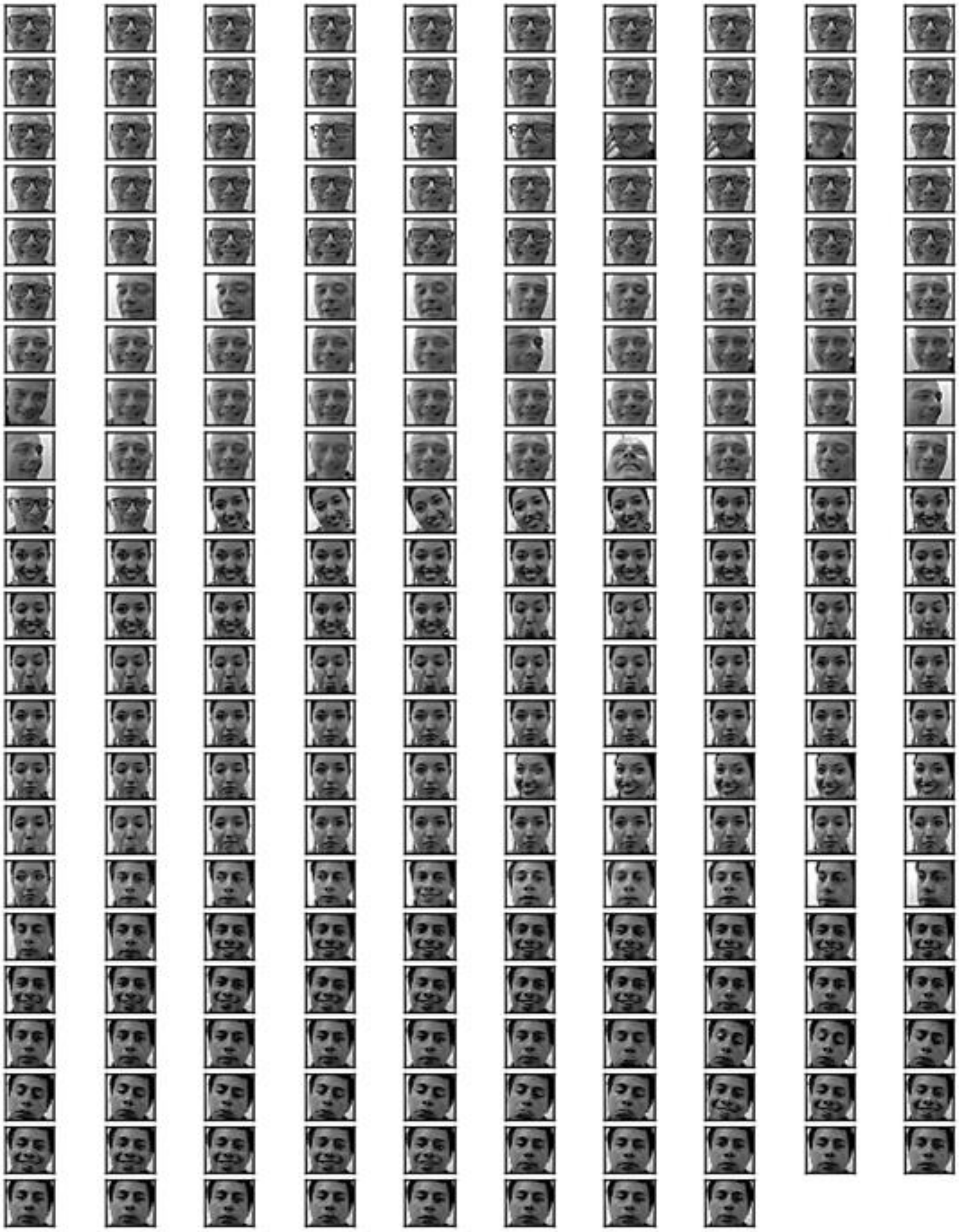


Figure 30. Data Set used to model the Face-Recognizing machine.

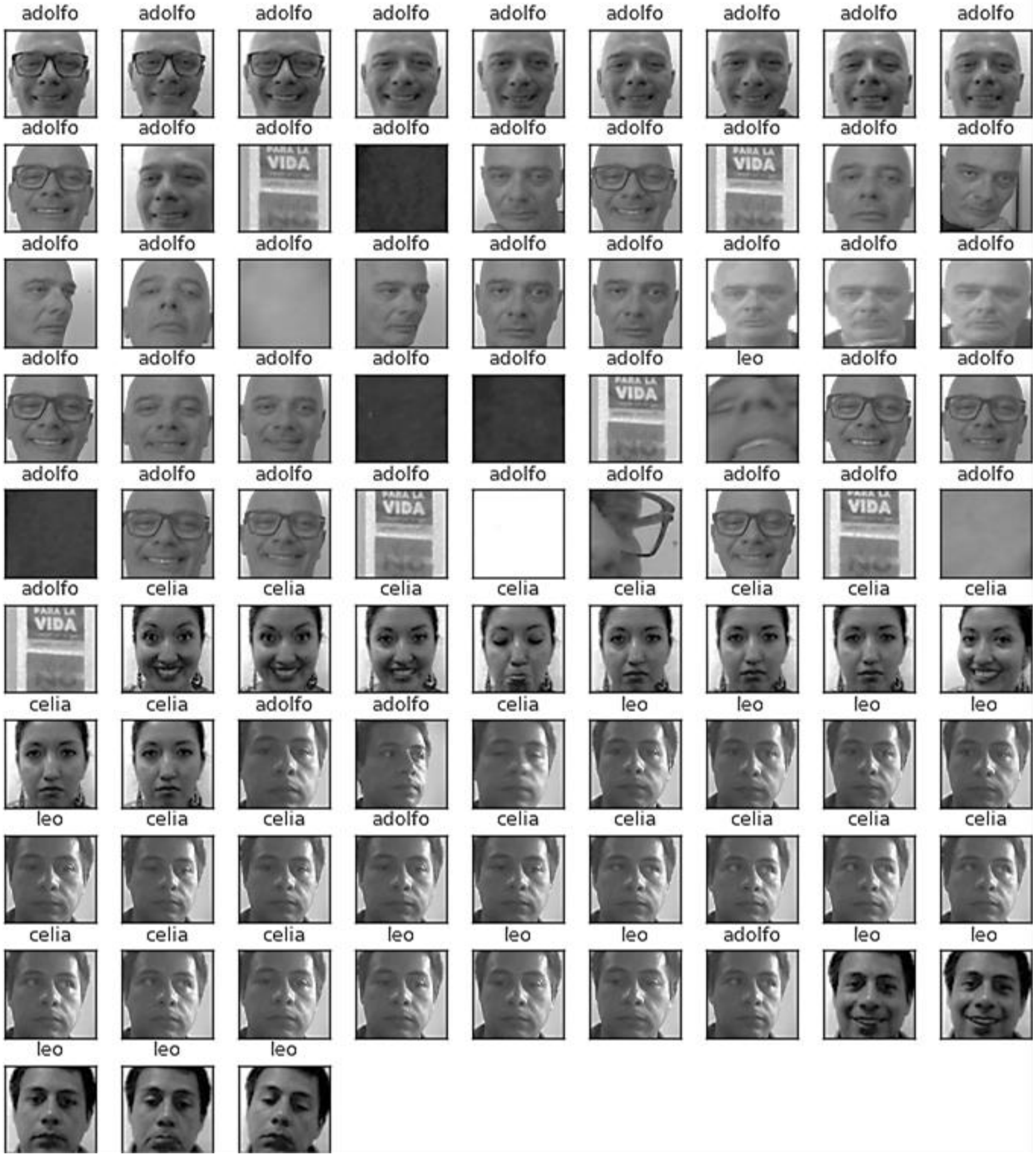

adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo
adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo
adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo
adolfo adolfo adolfo adolfo adolfo adolfo leo adolfo adolfo
adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo adolfo
adolfo celia celia celia celia celia celia celia celia
celia celia adolfo adolfo celia leo leo leo leo
leo celia celia adolfo celia celia celia celia celia
celia celia celia leo leo leo adolfo leo leo
leo leo leo

Figure 31. Test Data Set used to test the Face-Recognizing machine.

Facial expression recognition.

Facial emotion perception plays an important role in computational intelligence as well as in the cognitive and neuroscience fields [66]. Facial expressions recognition not necessarily allows to perceive human emotions, but it could be helpful to understand the human opinions and attitudes [67] or behaviors [5]; thus, facial expressions recognition constitutes a powerful tool that would greatly enhance the HRI [58].

Facial expressions have a systematic, coherent and meaningful structure that can be mapped into affective dimensions, that is, they can be measured in one way or another [52]. The **Facial Action Coding System** (FACS) [3] is the most common method used for recognizing emotions [68]. FACS analyzes and codes the facial muscular activity as **Action Units** (AUs) [69]. AUs combination can be used to distinguish between facial emotions. However, other methods such as 2D (two dimensional) and/or 3D (three dimensional) face shape analysis has emerged in most recent years [70]. Facial features can be described by geometric shapes extracted from a set of points [71] that represent parts or regions of the face (eyes, mouth, etc.). All these methods generally are complemented with Support Vector Classifier (SVC) or Linear Discriminant Analysis (LDA) for classification tasks [72].

The six basic facial expressions of emotion are anger, happiness, fear, surprise, disgust and sadness [3]. Unfortunately, the NAO Robot does not provide an API module to perform facial expressions recognition, the most relevant information that NAO Robot can provide is the 31 facial mark point which can be analyzed to carry out the facial expressions recognition successfully.

Using the same criteria as the Face recognition detailed in the previous section, a Facial Expression Recognizer was developed in order to provide to the NAO Robot the ability to interpret emotions and react to them. The expressions considered were: happiness, sadness and neutral; these three facial expressions have higher facial recognition rates than other emotions such as fear, angry and disgust. Neutral is not a basic facial expression but it was used as reference and surprise facial expression will be analyzed later. Unlike other procedures, in this case, the images were taken from the same person; however, in order to ensure greater accuracy, facial expressions were simulated by a theater actor (Figure 32).

As the results show (Figure 33), successful recognition rates for the facial emotions is lower compared to Wolfram Mathematica based classifier (Appendix B); however, face detection efficiency is better, therefore both have same performance (Table 7). Recognition rates of Wolfram Mathematica based and Python based expression recognizers.

Table 7. Recognition rates of Wolfram Mathematica based and Python based facial expression recognizers.

Classifier type	***Face***	***Happy***	***Sad***	***Neutral***	***Total***
Mathematica	54%	92%	85%	90%	80%
Python	71%	83%	71%	86%	77%

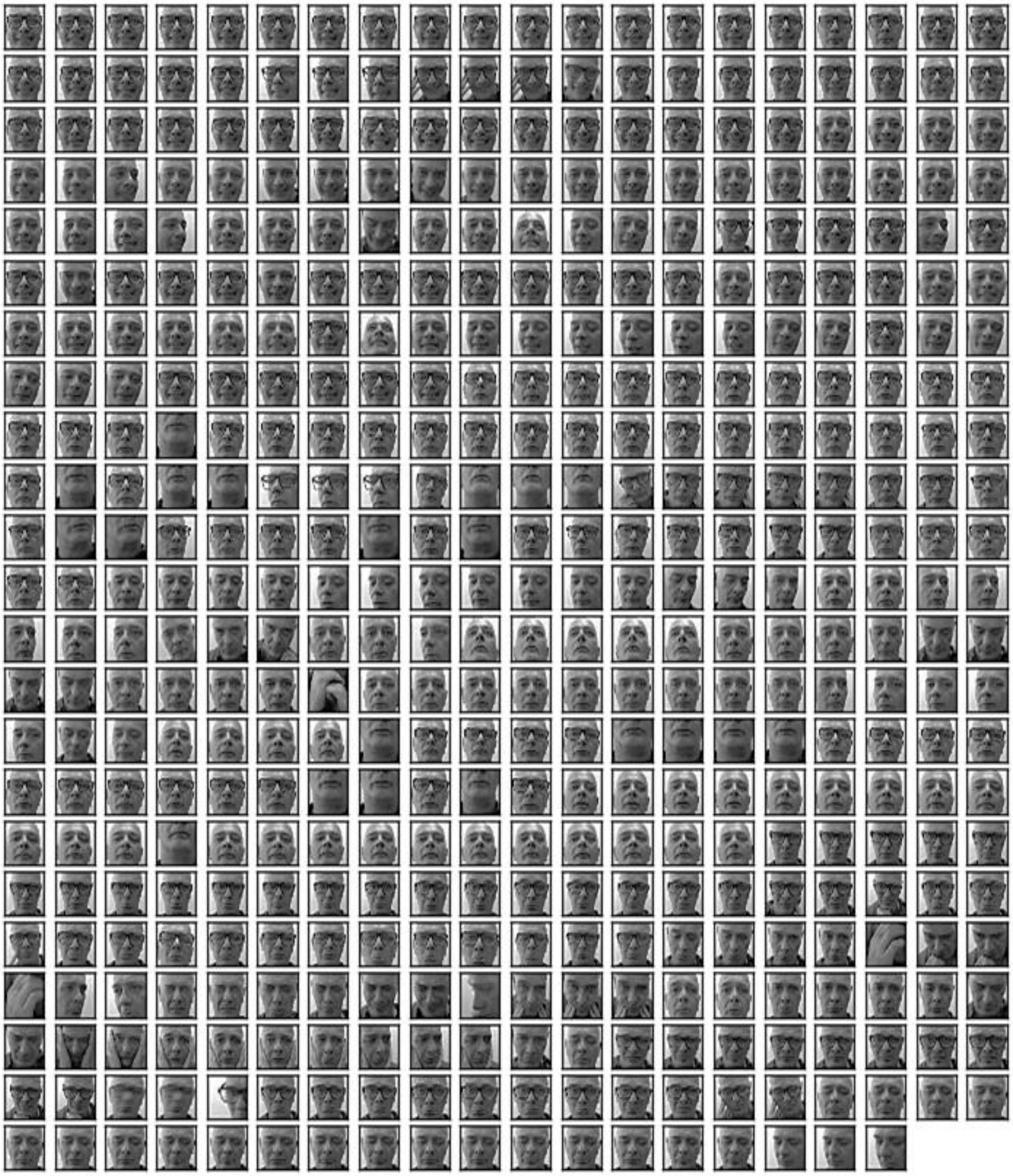


Figure 32. Facial expressions of Happy, Neutral and Sad used to model the Expression recognizermachine. Images were represented by an theater actor.

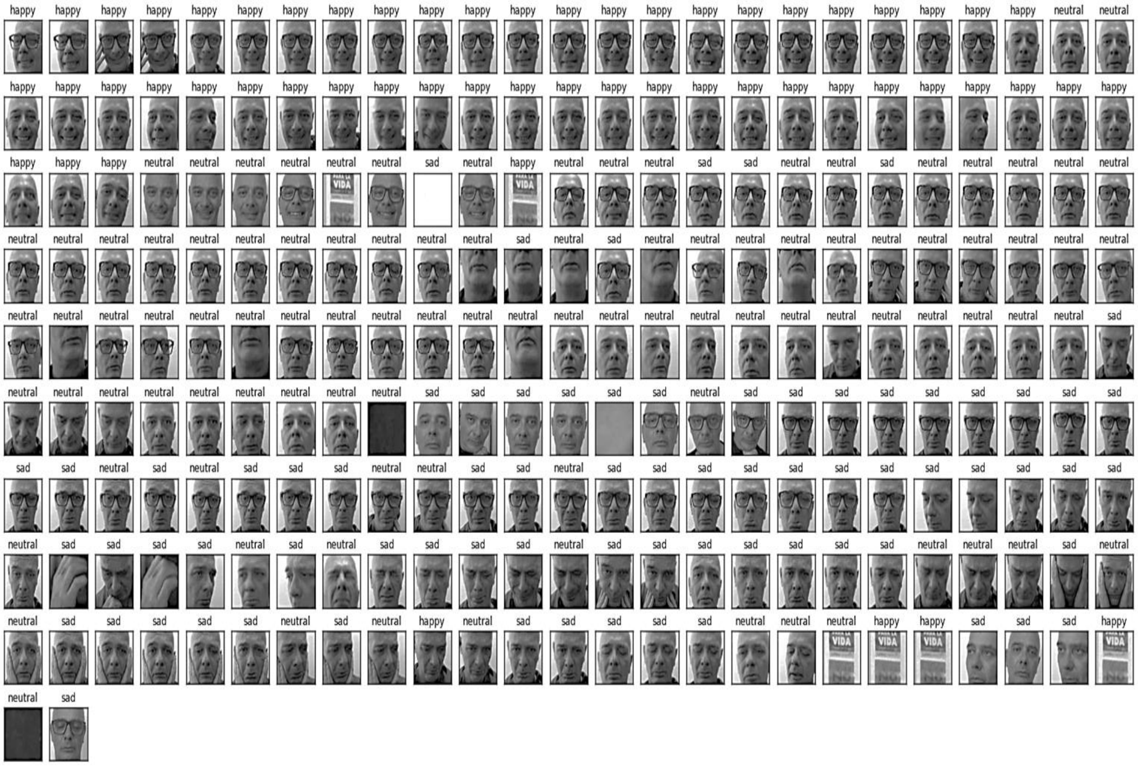


Figure 33. Facial expressions test Data Set used to test Facial expression recognizer machine.

As mentioned previously , the NAO Robot generates 31 points for the face representation (Figure 34). Each point is represented by a pair of x and y coordinates [58]; whit this information, it is possible to reduce the computational cost (if trying within the NAO Robot) by adopting a simple geometric approach in order to recognize facial emotions.

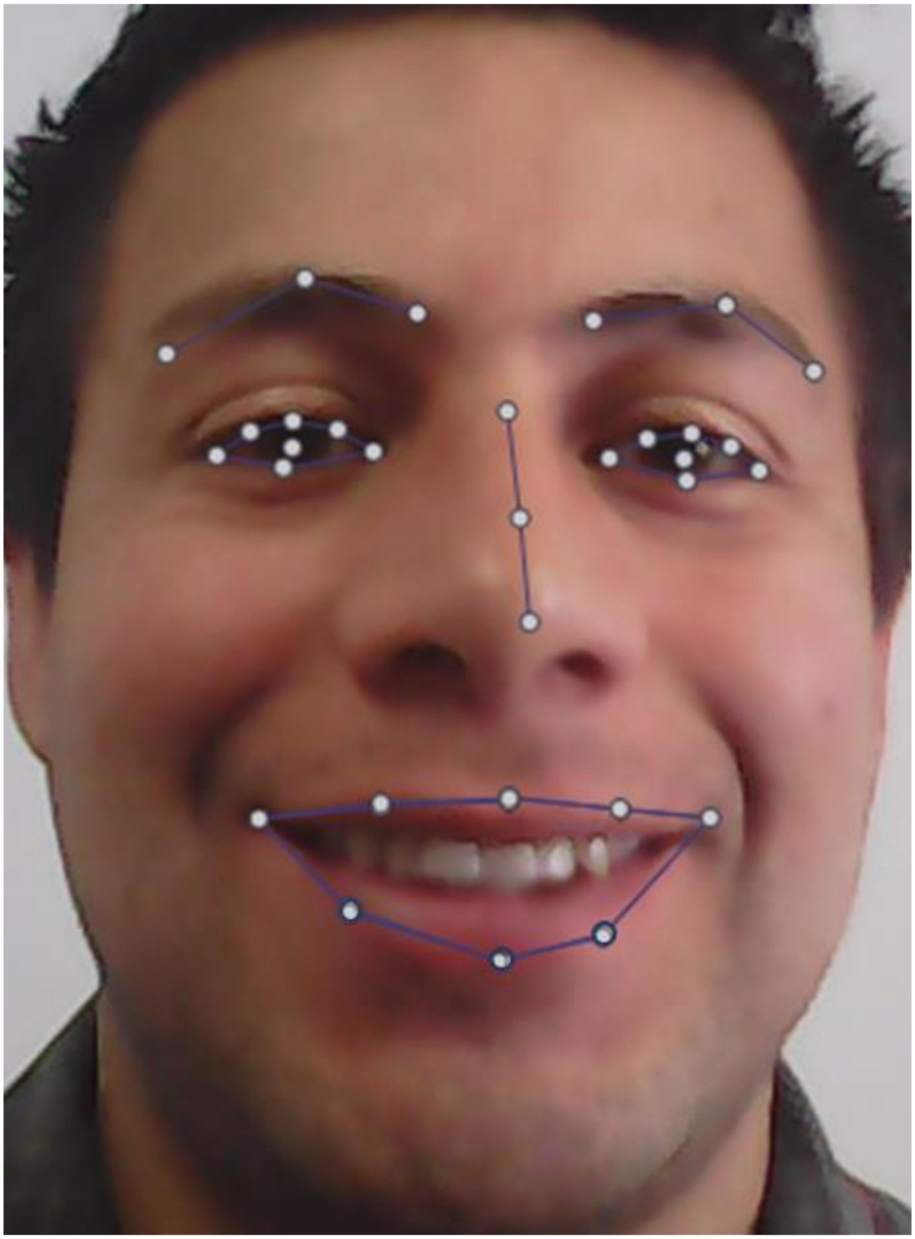


Figure 34. Face Land Marks provided by NAO's ALFaceDetection module.

In order to increase the facial expression recognition rate, a geometric analysis of the contours of the mouth was carried out. For this, a discriminant characteristic for each of the facial expressions is identified. The first factor to analyze is the shape formed by the outline of the mouth, When surprise is expressed, the mouth opens and the shaped figure resembles a circle (Figure 35, box 1), while for the other expressions (neutral, happiness and sadness), the figure is more similar to an ellipse (Figure 35, box 2, neutral & saddnes, and box 3, happiness).

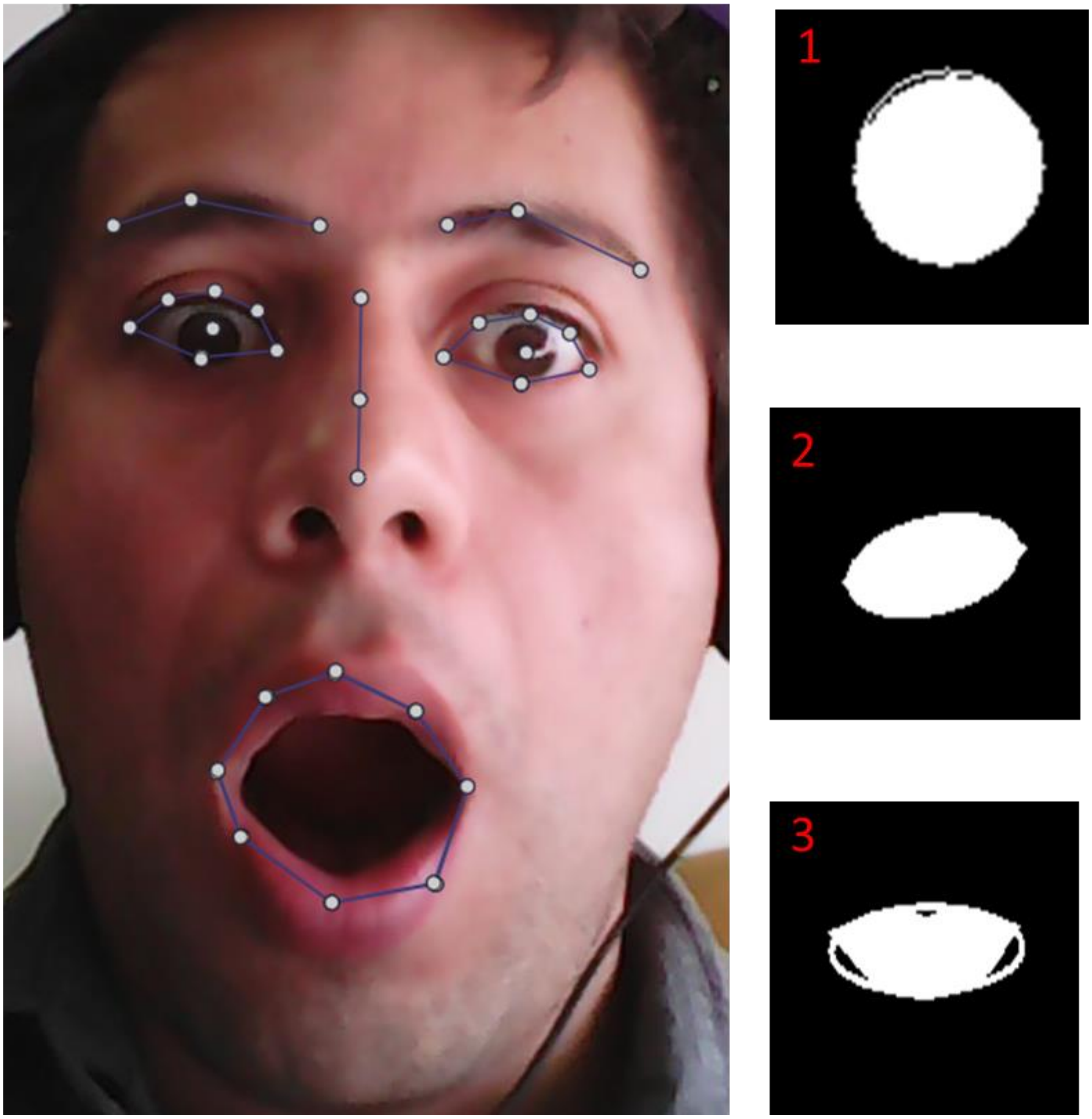


Figure 35. Mouth shape analysis: (1) Surprise, (2) Neutral-Sad and (3) Happy (video link 1- 2).

The eccentricity (e) is a mathematical parameter used to measure how much a conic shape varies from being circular. (Figure 36). So, if e <= 0.5, it is most likely that the expressed emotion is surprise.

The second discriminate factor used is based on the facial proportions. If we consider the facial symmetry of a neutral face [73]–[75], taking 4 of the 8 bilateral points (Figure 37) used in the estimation of horizontal and vertical asymmetry [73], it is possible to obtain enough information to determine if the mouth is featuring a smile; that is, expressing happiness.

$$\boldsymbol{e} = \sqrt{\boldsymbol{1} - \left(\frac{\boldsymbol{b}}{\boldsymbol{a}}\right)^{\boldsymbol{2}}}$$

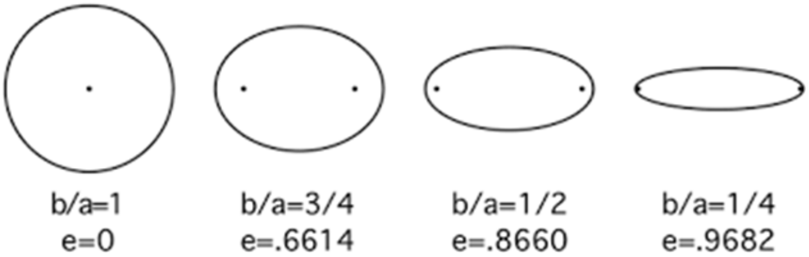


Figure 36. Analysis of the shape eccentricity. The eccentricity of a circle is zero, while the eccentricity of an ellipse is greater than zero and less than 1.

When the facial expressions are neutral or sad, D5 decreases, then D2 / D5 tends to the maximum value (assumed when D5 = D2), whereas D5 / D1 decreases (Figure 38, box 1). On the other hand, when the expression of happiness is manifested, the corners of the lips rise diagonally [76], making D5 to increase; thus, D5 / D1 tends to the maximum value (assumed when D5 = D1), while D2 / D5 decreases (Figure 38, box 2).

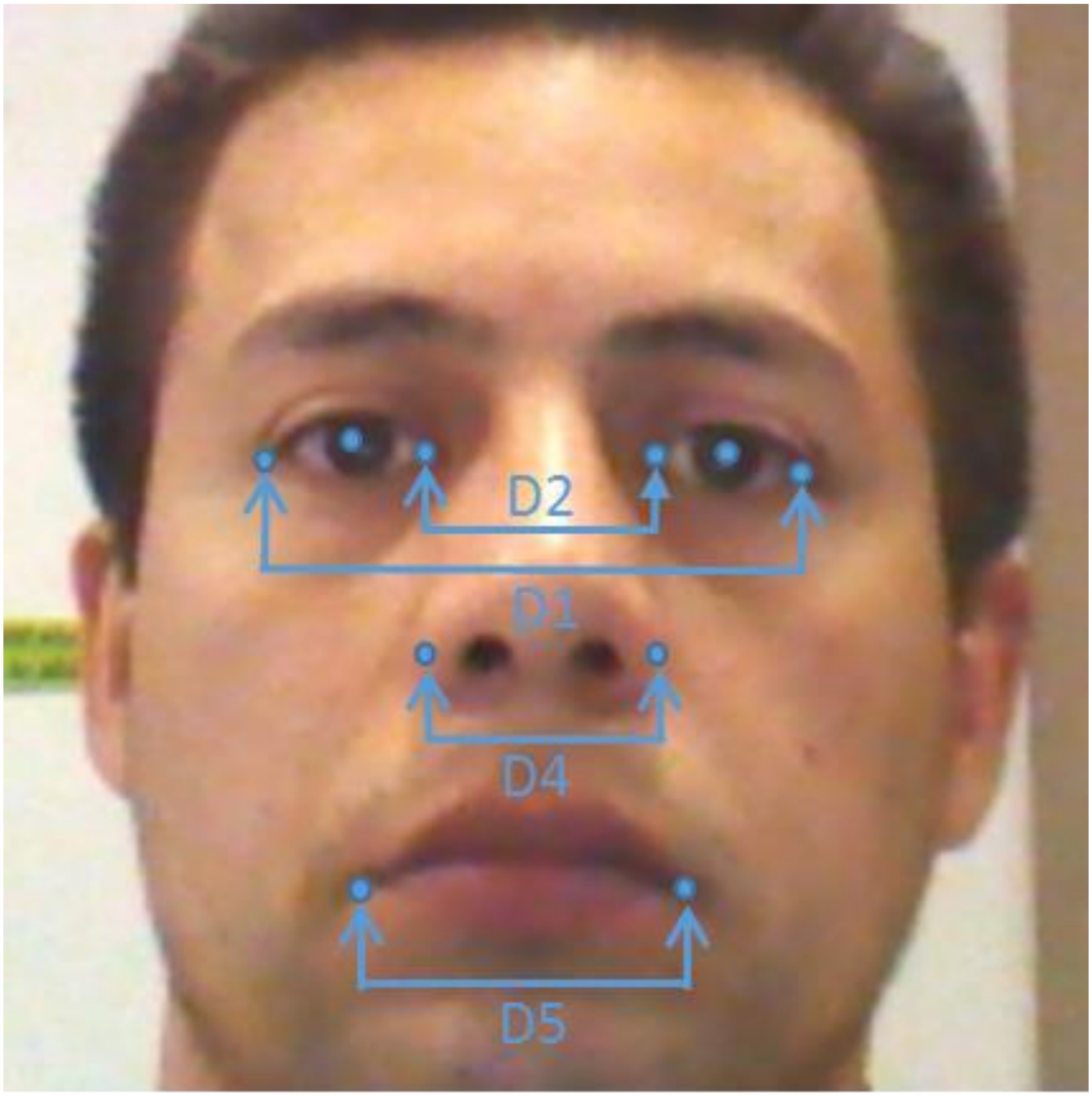


Figure 37. Bilateral point used for the calculation of facial proportions metrics.

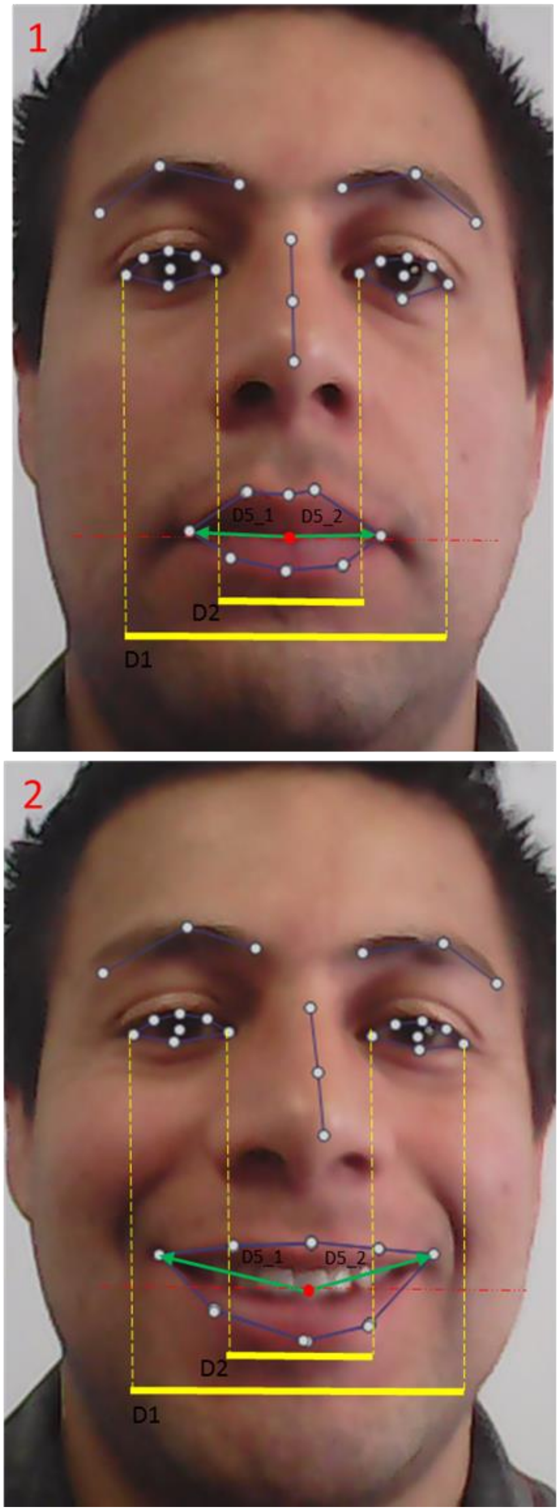


Figure 38. *Mouth-eyes symmetry ratios: (1) Neutral and (2) Happy. When happy is present, lip corners are raised diagonally.*

If D5 is the distance between right-left mouth extreme points and the center of the mouth, it is possible to generalize the discriminant feature as follow:

$$H = \frac{D5}{D1} - \frac{D2}{D5}$$

$$N = \frac{D2}{D5} - \frac{D5}{D1}$$

$$D5 = D5_1 + D5_2$$

$$Emotion = \begin{cases} Neutral\ or\ Sad\ , & H < N \\ Happy, & H \geq N \end{cases} \qquad (1)$$

The last step is to differentiate between neutral and sad facial expression. This is perhaps the most difficult task to perform, in some cases people show a facial expression that does not necessarily represent the current state or emotion; so if distinguish these two facial expressions can be challenging for humans, it is even more difficult for a robot

When sadness is manifested (Figure 39), lip corners are pulled down [76] and the shape created by the union of the points NC-ML-MB-ML seems a triangle (Figure 39,box 1); instead, the shape seems a quadrilateral when happiness or a neutral face is shown (Figure 39, box 3 and 2 respectively). To determine the figure, the contour shape is approximated to shape with less number of vertices depending upon the specified precision (3 for sad and 4 for neutral & happy). This can be done easily thanks to the OPENCV function "**ApproxPolyDP**" [59]. The accuracy parameter epsilon used is 0.02 * arcLength (shape perimeter).

The real-time test (Figure 40) gives an idea how this module will work when will used on the real robot.

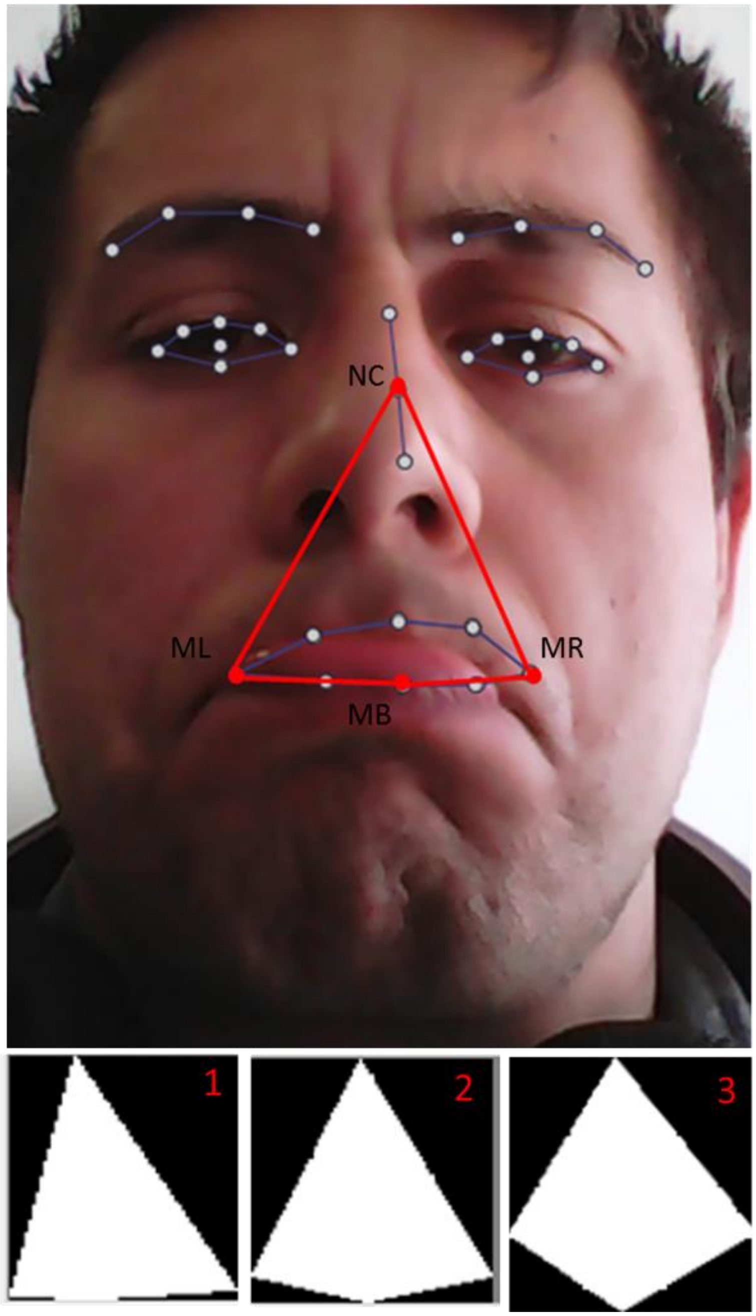


Figure 39. Mouth-nose shape analysis. (1) Sadness, the figure is a triangle, (2) Neutral and (3) Happiness, the figure is a quadrilateral (video link).

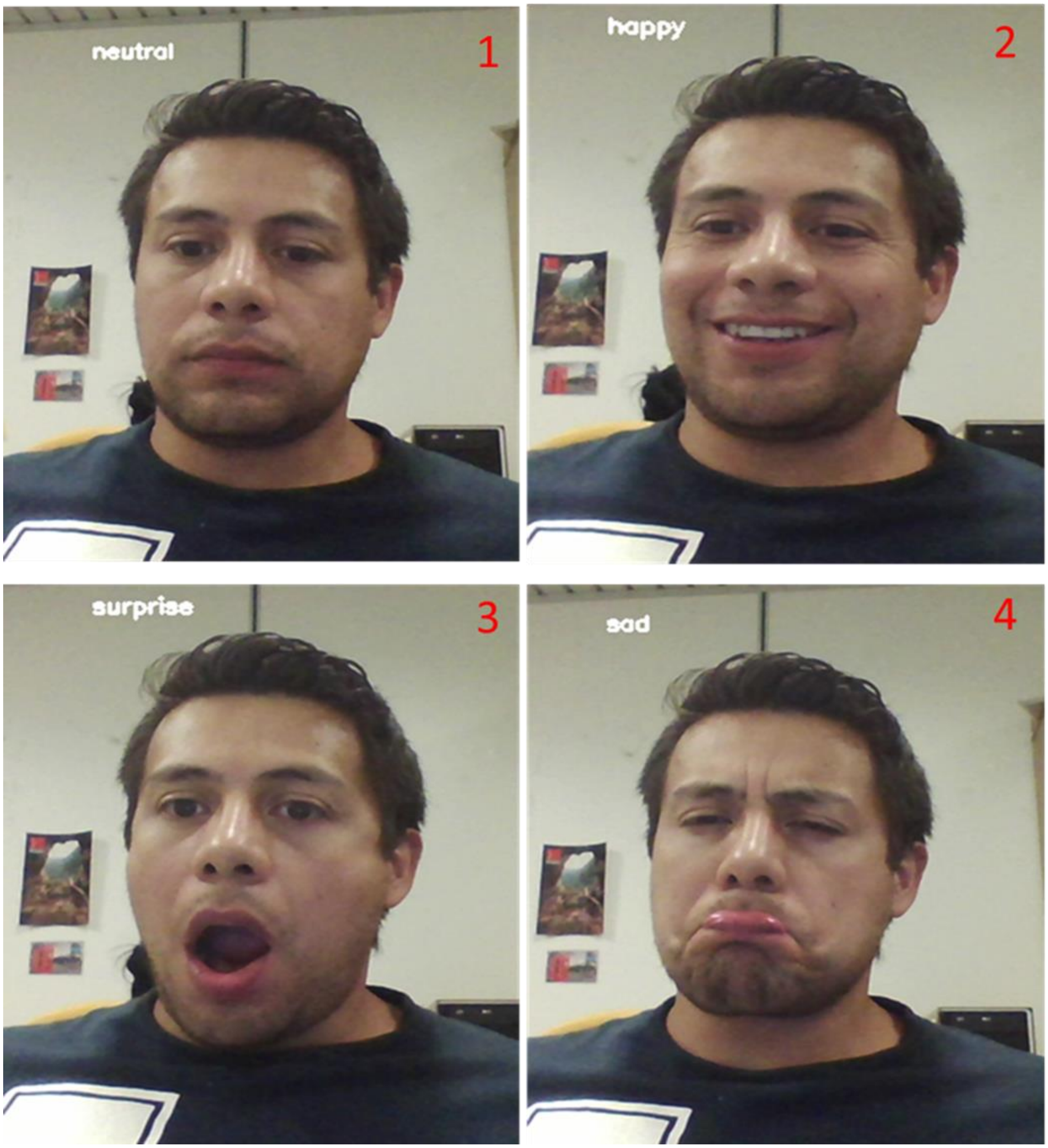


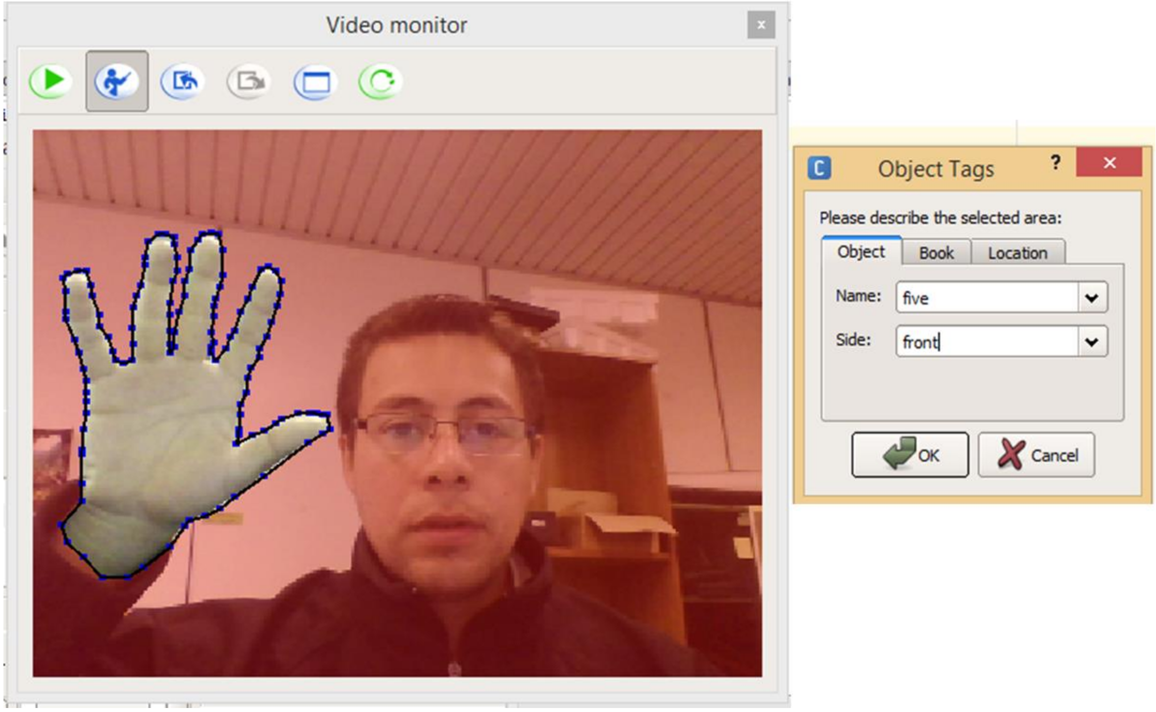
Figure 40. Emotions Recognizer real time results. (1) Neutral, (2) Happy, (3) Surprise and (4) Sad.

Hand Gesture Recognition.

Human gestures constitute a series of motion expressed using the body face and/or hands [77]. Gestures are used independently of cultural and linguistic origins and emerge in baby children even before language development [78]. Recent research has suggested that over 60% of communication is body language or nonverbal [79].

Gestures and body movement play an important role in human communicative behavior, and these gestures can be directly related to the information flow of interactions [80]. Recognizing gestures is not a simple task; it involves many aspects such as motion modeling and analysis, pattern recognition and machine learning. [81].

The NAO Robot APIs provide a static object learning and recognition module which can be used to detect a pre-learnt object with its cameras [79]. It can be done by using the Video monitor panel included in the Aldebaran software package [8]. Once an object image is outlined, the information is sent to the detection database (Figure 41); the NAO Robot is able to detect similar objects from its video stream. However, the detection accuracy highly depends on the environment illumination and background color; moreover, this cannot be tested on a simulated robot unless it is done in a virtual world (WEBOTS simulator).



Please describe the selected area:

Name: five

Side: front

Figure 41. *NAO's object recognition panel. Once an object image is outlined, the information is sent to the detection database. Object recognition feature works only within real NAO.*

To provide the possibility of controlling the robot with the hands, a gesture classifier machine was modeled. To reduce the complexity, it has been defined only five base gestures corresponding to the numbers from zero to five (Figure 42) that the NAO Robot may be able to recognize. The classifier was trained with a set of images corresponding to each base gesture. The sequential combination and defined movements, allow to generate new gestures.

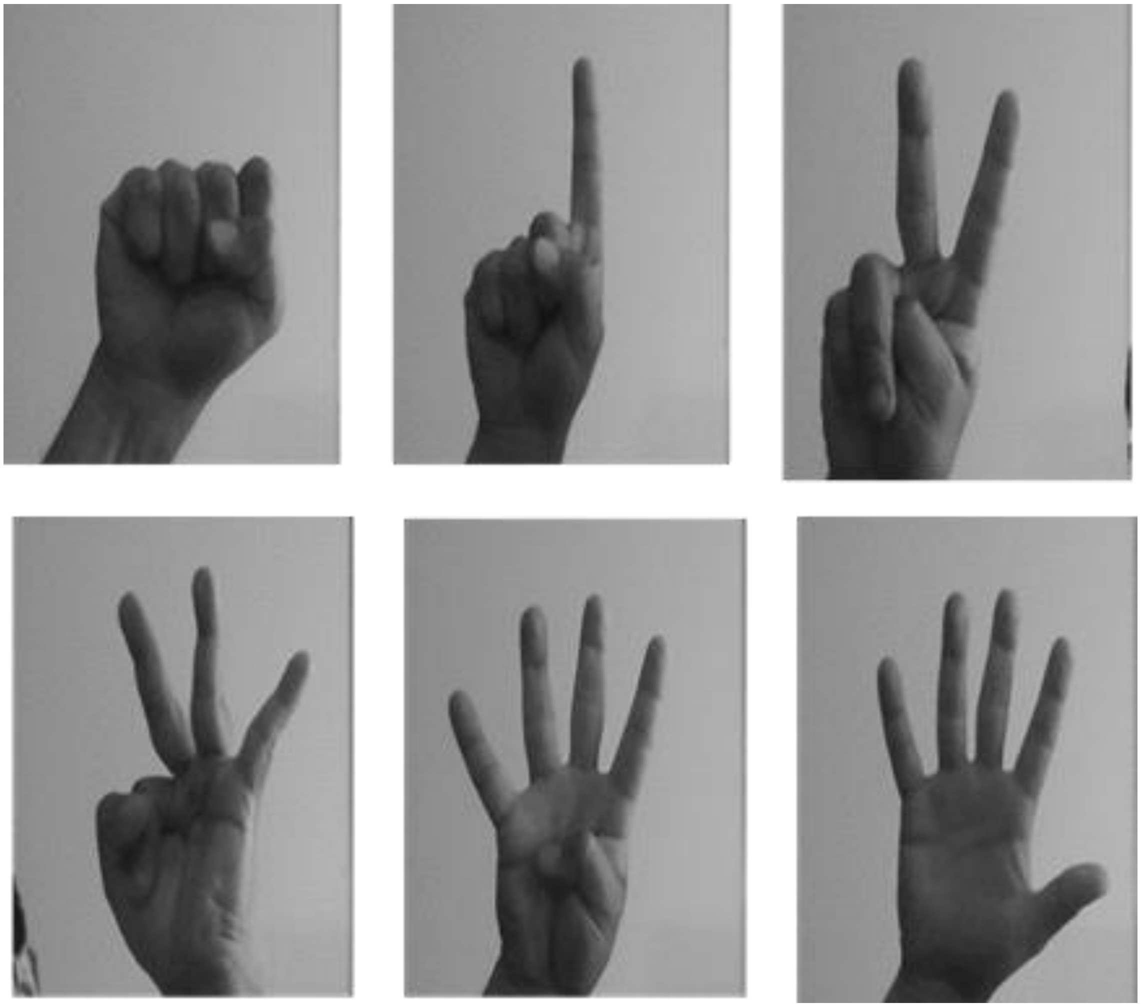


Figure 42. Hand Fixed Gestures. These represent numbers from 0 to 5.

The gesture classification process begins when a person is detected, similar to the face detection, a “Haar feature-based cascade” classifier is used, in this case, to detect the upper part of the body (**Body UP)** instead of a face (Figure 43, box 1). Then the position of the person with respect to the center of the image is determined (Figure 43, box 2). If the person is more to the left, the half-right of the image is analyzed or vice versa.

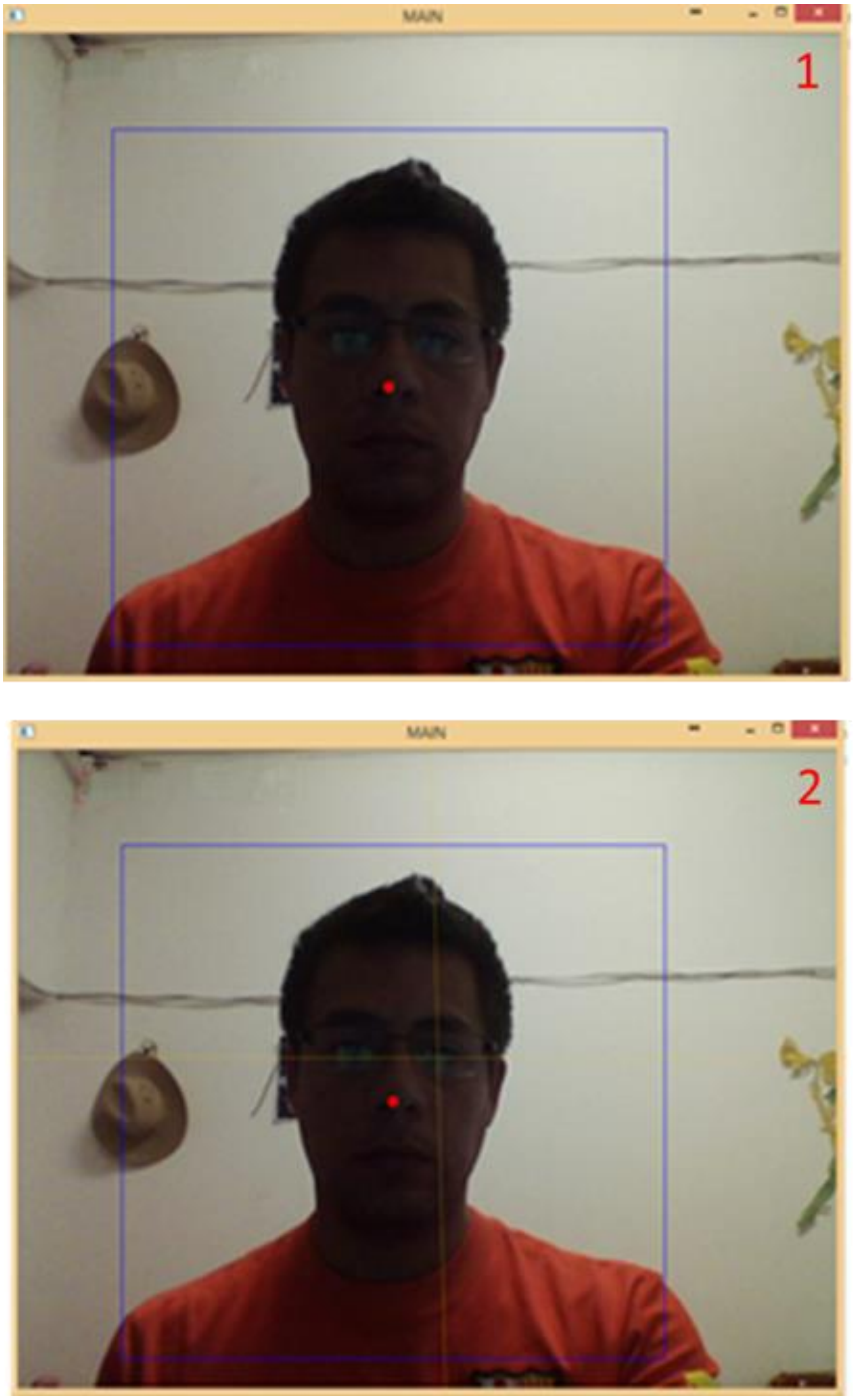


Figure 43. Body up Detection using OpenCV Haar feature-based cascade classifier. *If the person is more to the left, the half-right of the image is analyzed or vice versa.*

To recognize the gestures, the hand should first be placed on the shoulders, approximately at the head height (Figure 44). In order to increase success possibility, a fast parallel analysis is carried up. The hand convexity defects and Hull are considered. OpenCV python APIs allow to perform this task easily.

Once the hand is identified (Figure 45, box 1), a threshold is applied to discriminate colors that not corresponding to skin (colors were defined with an assessment type test and error). Then; by analyzing contours and areas, it is possible to detect hand contours (we assume that the hand has the greatest area) (Figure 45, box 2).

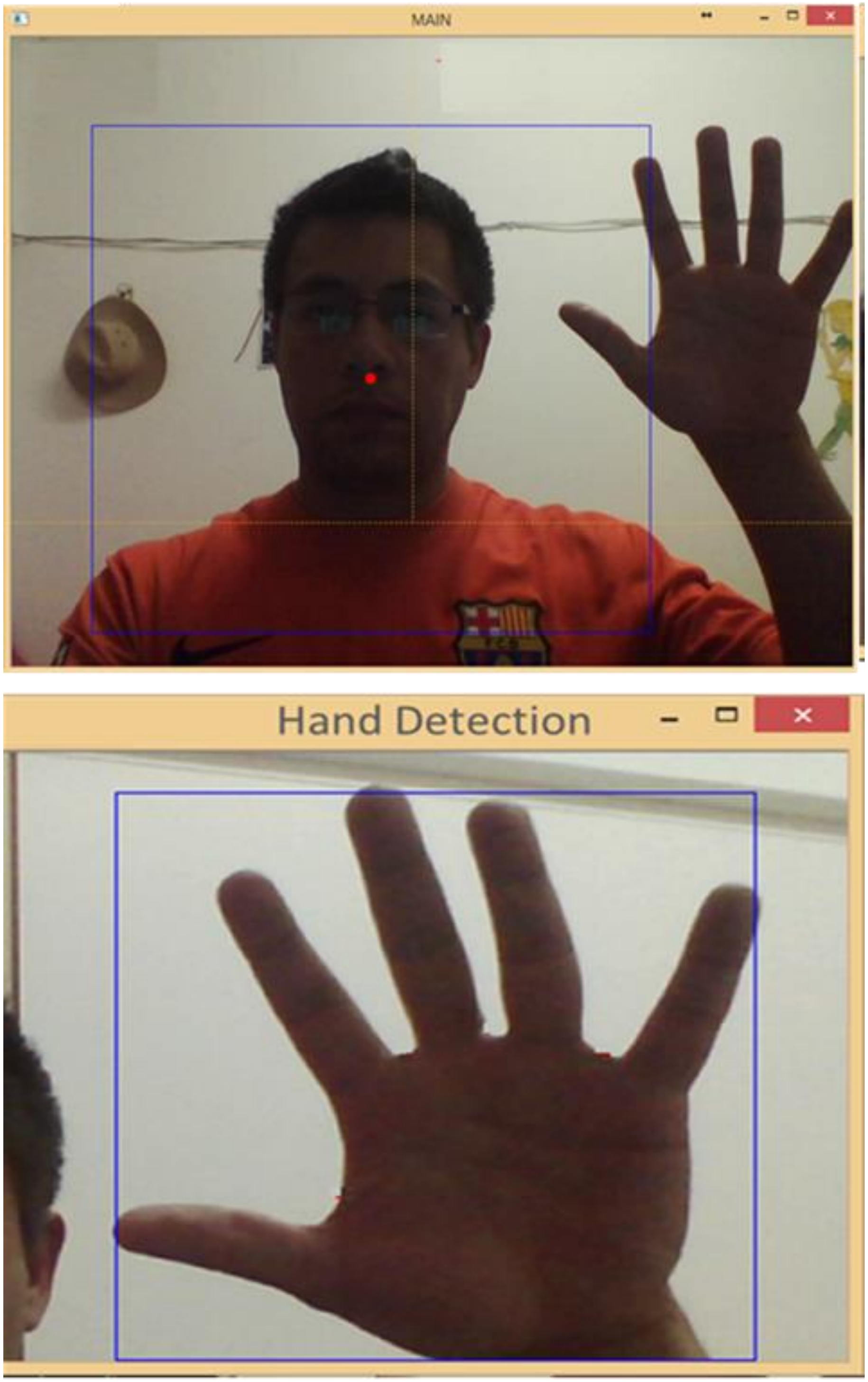


Figure 44. Hand position control. The ideal position of the hand is at the head height.

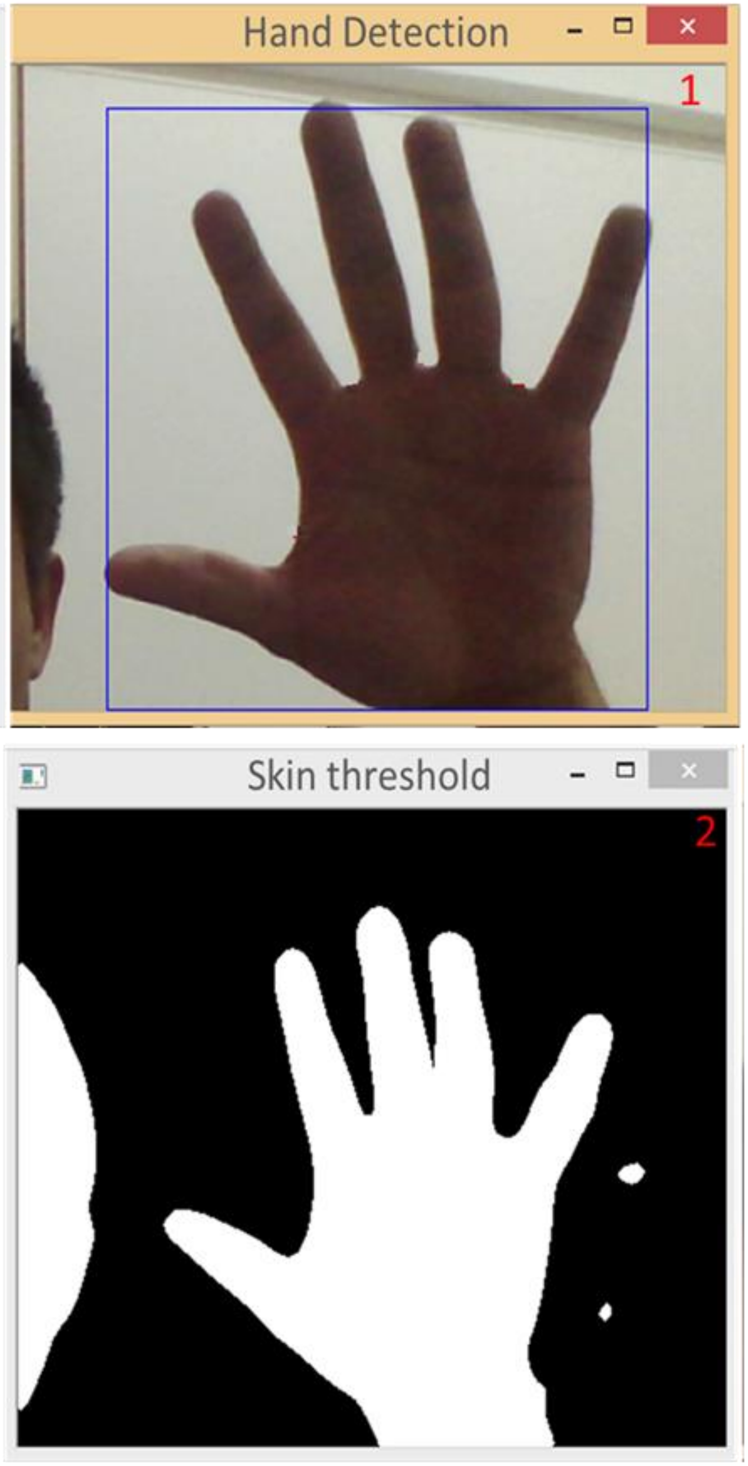


Figure 45. *Hand detection. Threshold is applied to discriminate colors that not corresponding to skin (colors were defined with an assessment type test and error).*

The hand convexity hull is defined by the extreme contours points in yellow (Figure 46). Convexity defects relative to that hull are regions limited by extreme points and defects points in red, which are the local maximum deviations of the hull from the contours. The defect angle in blue can be used as classification parameter. These convexity defects offer a mechanism to characterize not only the hand itself, but also its state (e.g. number of open fingers) [82].

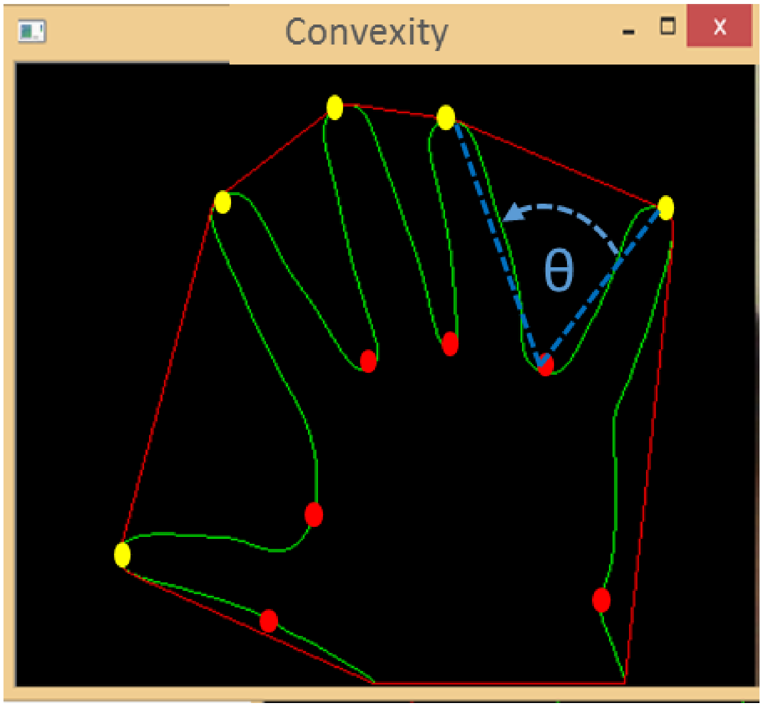


Figure 46. Hand convexity defects and hull. Extreme contours points in yellow. Convexity defects in red.

Consider that a finger is open when θ is greater than 0 ° and less than 90 ° (Figure 47), the number of open fingers can be defined as N + 1, where N represents the defects that satisfy the condition. A special case occurs when N = 0, here the number of open fingers could be 0 or 1 and it will be defined directly by the classifying machine. Additionally, the NAO Robot examines the sonar sensors to assist the gesture recognition. When it detects a person within a range of 1.5 meters, the sonar sensors will trigger its gesture recognition system (Figure 48).

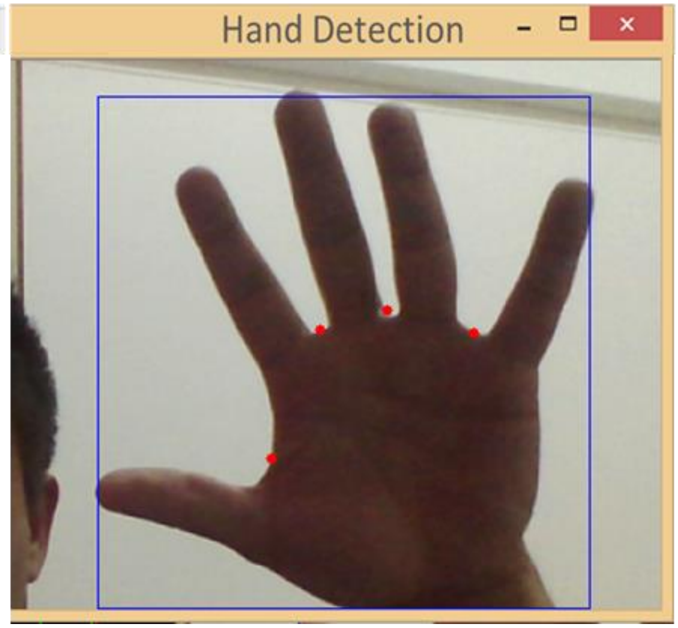


Figure 47. Fingers count. A finger in open when θ is grater 0° and less than 90°. the number of open fingers can be defined as N + 1, where N represents the defects that satisfy the condition (red points).

The previous method described above considers only one hand, either right or left hand depending on the person position; however, it could be extended to the use of both hands when there is an appropriate distance (controlled by NAO Robot's ultrasonic sensors) and position between robot and person.

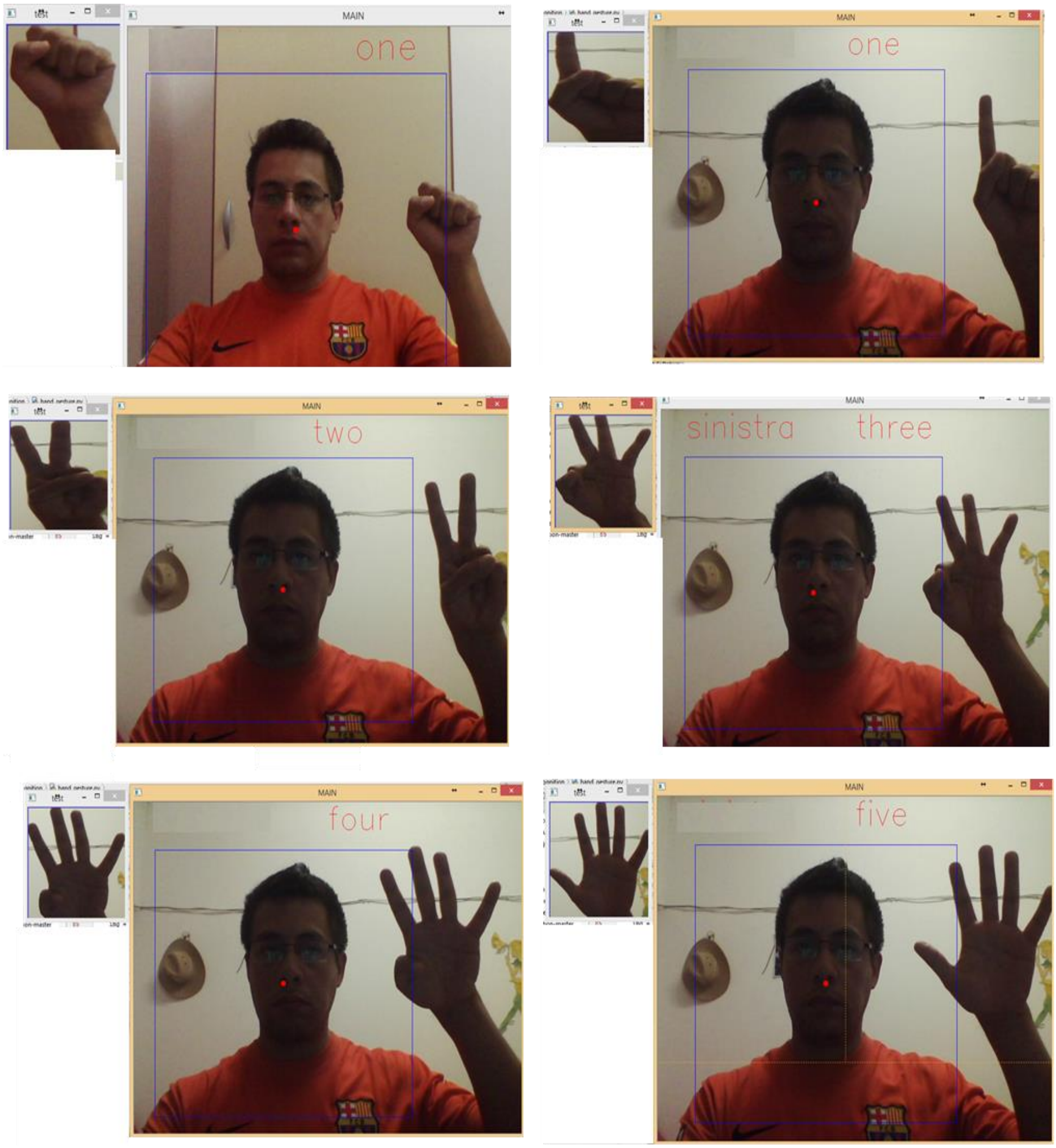


Figure 48. Gesture recognition test using one hand. Most of false positives occurred when recognizing the 0 and 1.

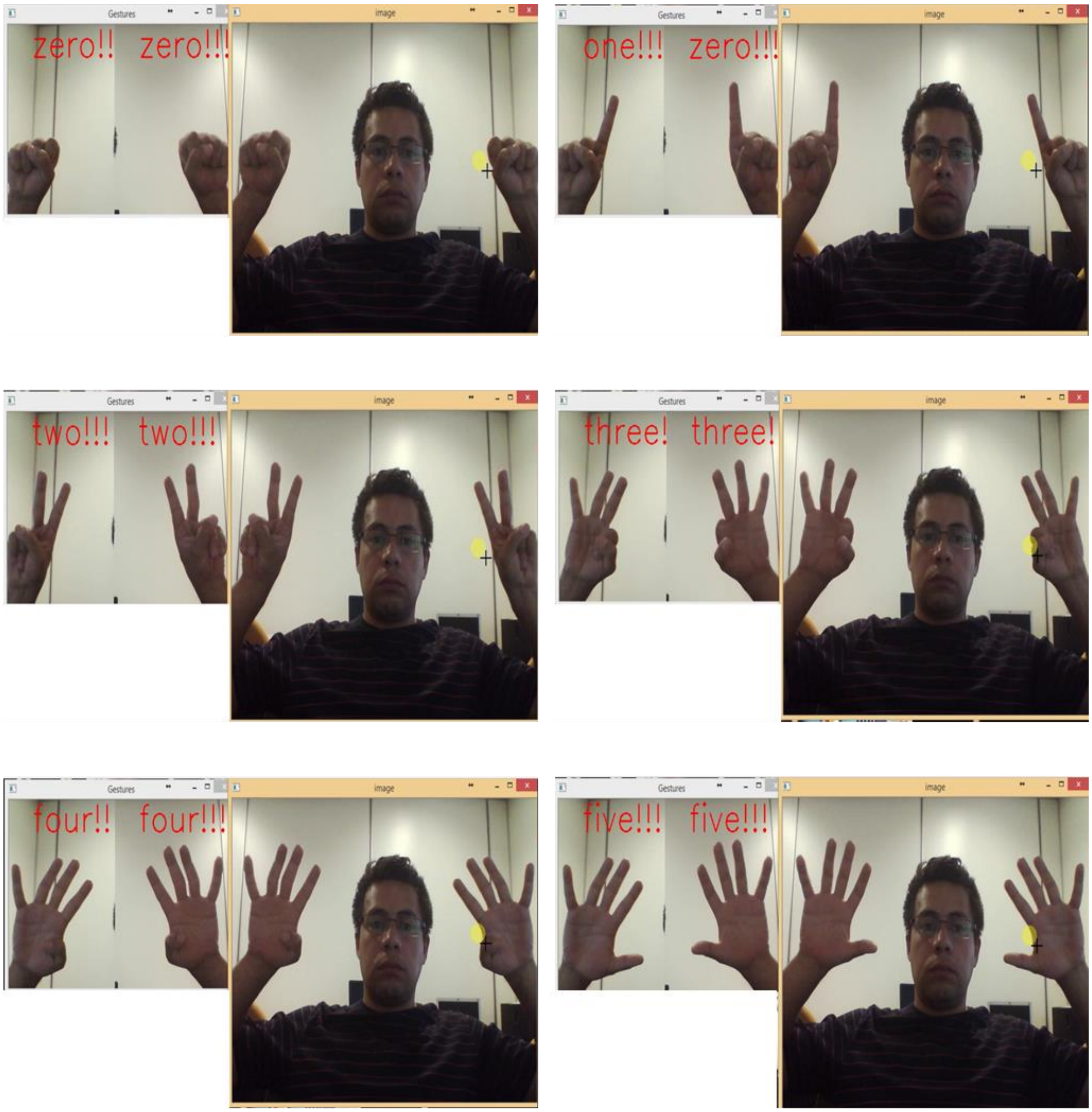


Figure 49. Gesture recognition test using both hands (video link).

NAO Robot Expresses Emotions

Humans can communicate verbally means by speech or nonverbally by facial expressions and body language [83]. Humans can express in a conscious or unconscious manner additional information using a variety of emotional and social gestures and signals [84].

In recent years, robots are being increasingly used in social environments, such as hospitals, schools or at home. When interacting with robots, people are mainly impressed by the robot's appearance. “Non-interactive” robots can provide support for physical rehabilitation and medication delivery tasks, whereas “highly interactive” robots can provide more cognitive, affective and social support [85].

NAO Robot does not have the ability to display facial expressions as human because of its plain face. However, the emotions can be expressed using arms and hands movements and playing sounds [86]. NAO Robot has the ability to alter its body posture and thus show affective postures; additionally, it can show different colors in its eyes and this can help to strengthen the emotions being expressed [85].

Some body language movements for NAO Robot were preprogrammed using Choregraphe software; it combines different LEDs colors and sound allowing the robot to express emotion. The patterns are mimic from human body language in real life. Emotions that can be portrayed using NAO's body postures are sadness, happiness and surprise (Figure 50) the same expressions that the robot can identify in facial expressions. The goal of this program is to allow the NAO Robot to express feelings through body language.

The NAO Robot can express happiness by clapping its hands, playing a baby’s laughing sound and turning its eyes green (Figure 50, box 2). To express sadness, the robot mimics the crying action, it plays a sound of a baby crying and turns its eyes red (Figure 50, box 3). Finally, to express surprise, the NAO Robot moves its head and chest backward, it puts it arms open with straight forearms, its hands open, plays a "WOW' sound and turns its eyes yellow (Figure 50, box 4).

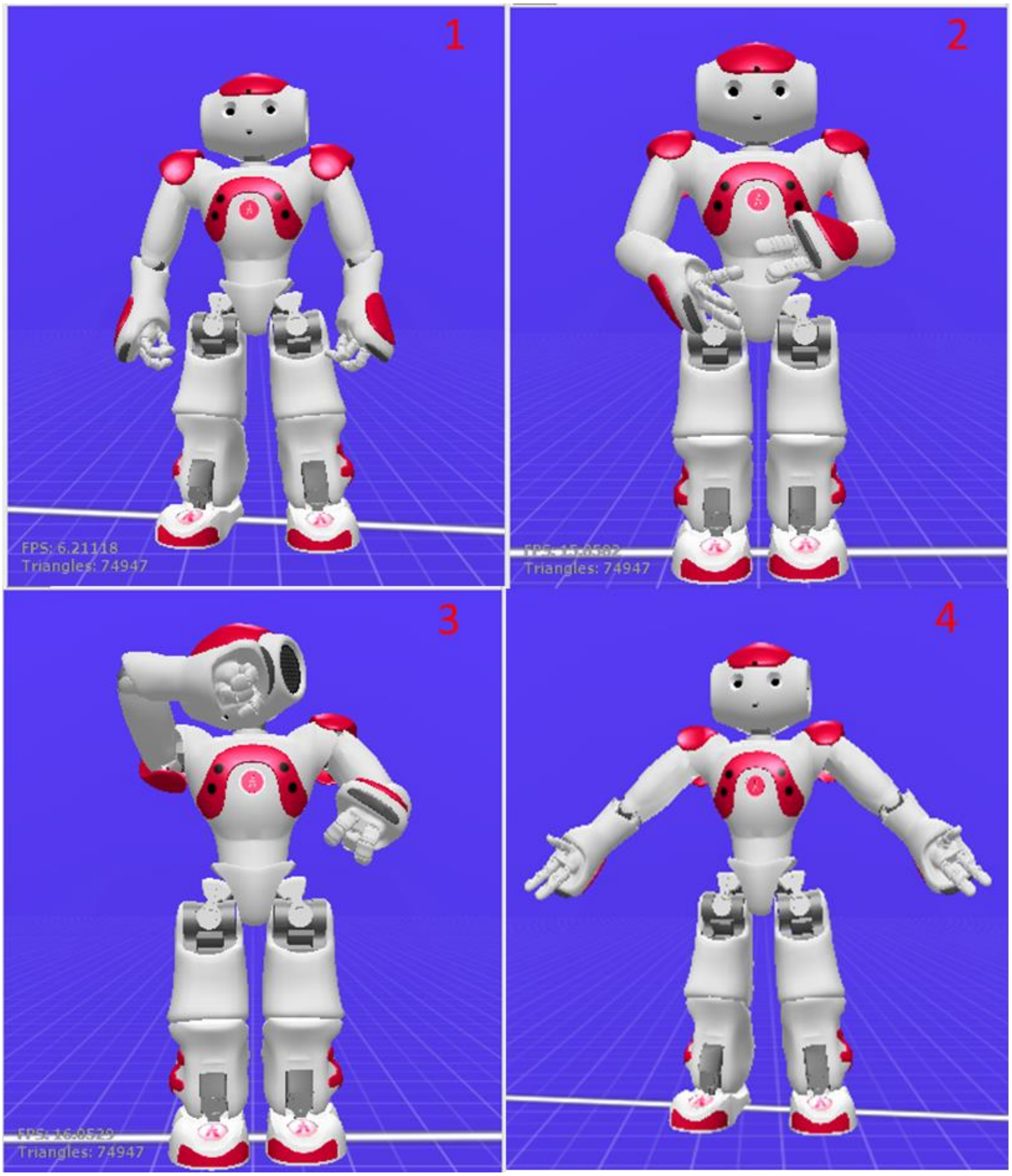


Figure 50. NAO Robot's postures for expressing emotions. Neutral (1), Happiness (2), Sadness (3), Surprise (4).

Chapter 5

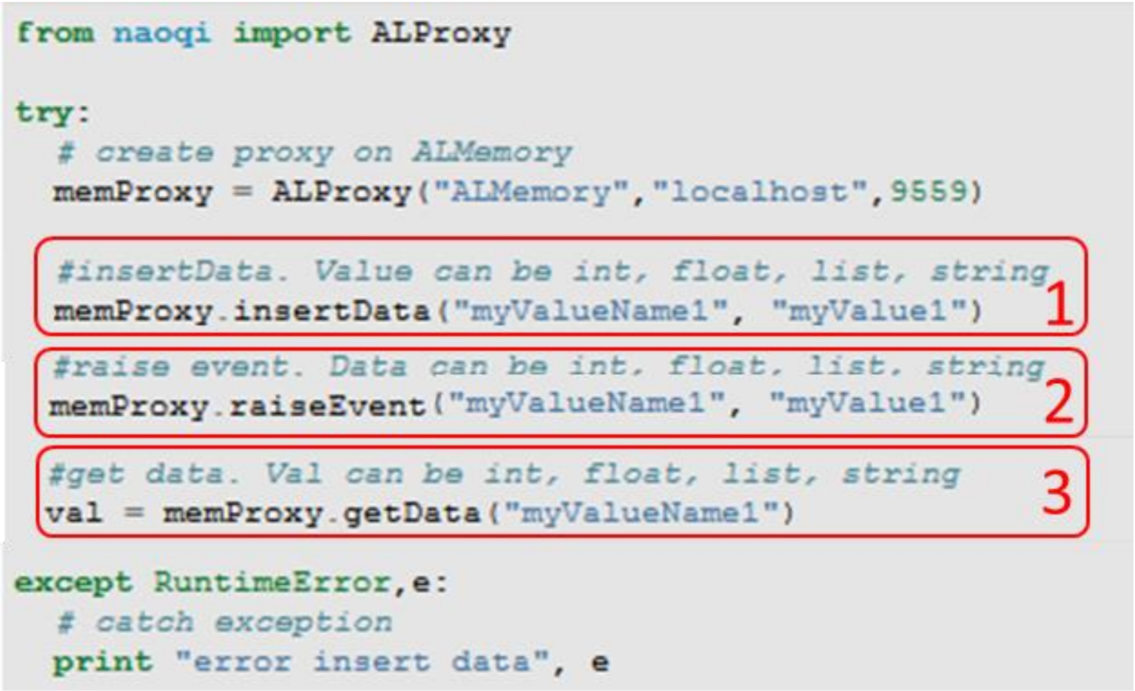
EXPERIMENTAL RESULTS

Until this moment, the NAO Robot is able recognize hand gestures, facial emotions and express feelings; however, each application controlled by a Python script was tested separately. The objective of this chapter is to embed these modules into the Aldebaran Choregraphe environment.

One way to directly execute python custom scripts within NAO's Choregraphe is by modifying a Box (or creating a new Script box) [8], in this way, all processing is performed by the NAO Robot CPU. In addition, all additional libraries must be installed in the real robot, since in the virtual or simulated is not possible.

An alternative way to embed external python modules within Choregraphe without spending excessively the resources of the NAO Robot CPU is through the "AlMemory" module available in the NAOqi. In this way, it is possible to remotely access all the robot variables with a simple read / write operation. The main advantages of this method are: the possibility of using a HOST with better CPU performance to process large amounts of data, and the possibility of using the simulated (or virtual) robot, which allows testing and / or changes quickly and safely.

The "ALMemory" module allows you to get or set any variables in the NAO Robot memory such as sensors, angles, postures, etc. It also gives you the ability to create customized variables [8]. A variable can be read using the Python command "**getData**", whereas the "**InsertData**" or "**RaiseEvent**" commands can be used to write within a variable (Figure 51).



```
from naoqi import ALProxy

try:
  # create proxy on ALMemory
  memProxy = ALProxy("ALMemory","localhost",9559)

  #insertData. Value can be int, float, list, string
  memProxy.insertData("myValueName1", "myValue1")
  #raise event. Data can be int, float, list, string
  memProxy.raiseEvent("myValueName1", "myValue1")
  #get data. Val can be int, float, list, string
  val = memProxy.getData("myValueName1")

except RuntimeError,e:
  # catch exception
  print "error insert data", e
```

Figure 51. NAO's memory access using python. (1) Writing a memory variable without handle an event. (2) Writing a memory variable with handled event. (3) Reading memory variable.

Unlike to the insert data (Figure 51, box 1), the rise event (Figure 51, box 2), in addition to changing the value of the variable creates an event that notifies when the event value (variable) changes. Both of them create a new variable if is not present on memory. The variable value could be read any moment (Figure 51, box 3); however, rise event could be handled and used with a callback function (interruption).

The interactive process begins with the “**Init”** sequence. Here, pre-trained classifier machines are loaded (Figure 52). Then, if someone is present and its face is detected, the Facial expression analysis starts; only if the expresion is different from Neutral, the event **"Emotion"** is raised. After that, the gesture analysis is launched, when a predefined gesture is detected, **“RightHand”** and/or **"LeftHand"** event is invoked, otherwise, the sequence restarts. Finally, into Choregraphe those events can be monitored easily by adding to the Flow diagram Space the corresponding memory events (Figure 53).

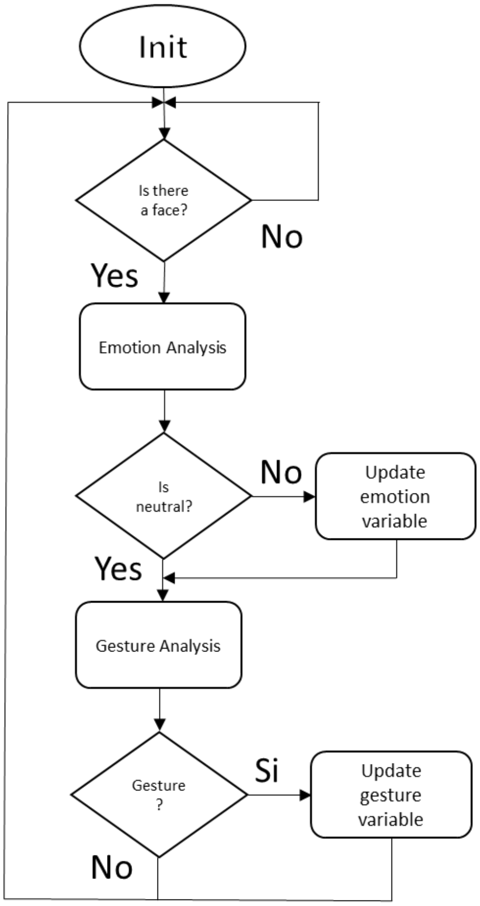
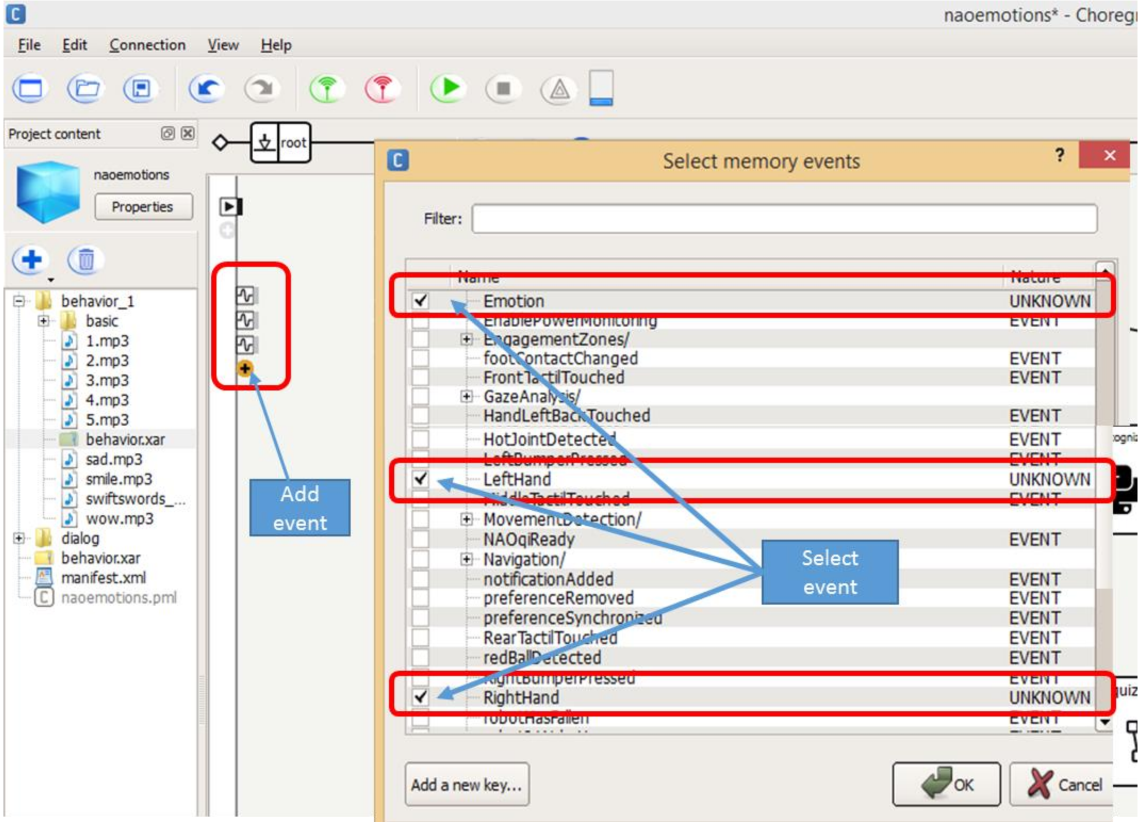


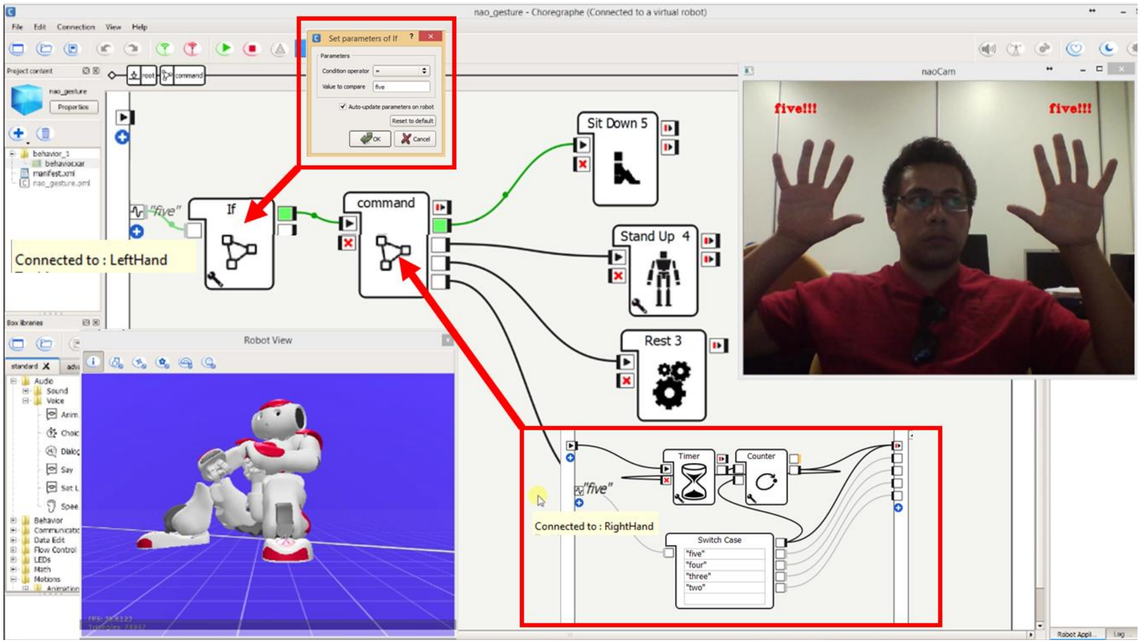
Figure 52. Flow diagram of the gestures and emotions controller module.



	Name	Nature
☑	Emotion	UNKNOWN
☐	EnablePowerMonitoring	EVENT
☐	⊞ EngagementZones/	
☐	footContactChanged	EVENT
☐	FrontTactilTouched	EVENT
☐	⊞ GazeAnalysis/	
☐	HandLeftBackTouched	EVENT
☐	HotJointDetected	EVENT
☐	LeftBumperPressed	EVENT
☑	LeftHand	UNKNOWN
☐	MiddleTactilTouched	EVENT
☐	⊞ MovementDetection/	
☐	NAOqiReady	EVENT
☐	⊞ Navigation/	
☐	notificationAdded	EVENT
☐	preferenceRemoved	EVENT
☐	preferenceSynchronized	EVENT
☐	RearTactilTouched	EVENT
☐	redBallDetected	EVENT
☐	RightBumperPressed	EVENT
☑	RightHand	UNKNOWN
☐	robotHasFallen	EVENT

Figure 53. Choregraphe memory events.

To check if script is running properly, a small application in Choregraphe has been used (Figure 54). The aim is to make move the robot using a combination of hand gestures. A command initiates when "five" number is detected on left hand, then, right hand indicates the movement; with number "five", "four"," three" and "two", for Sit down, Stand up, Rest and Wakeup commands respectively. Numbers "one” and "zero" were not considered because of a lot of false positives.



Set parameters of If

Parameters

Condition operator: =

Value to compare: five

☑ Auto-update parameters on robot

Reset to default

OK Cancel

Figure 54. Choregraphe test program. The NAO Robot is controlled using hands (video link).

A complete test was developed during the Researchers' Night event at UNICAL University. Here, high school students from different local institutions participated in the activity **"Programmare giocando"** (program by playing) developed by the Department of Physics of the UNICAL university. As a part of this activity, students interacted with the NAO Robot.

The NAO Robot was programmed to perform three specific tasks. For the first one "**Vuoi giocare con me?**" (Would you like to play with me?), a test of Italian general culture was the first contact between the students and the robot (Figure 55). In this activity, the students had to answer some questions asked by the robot; the robot assessed whether the answers were correct or not. While the NAO Robot was speaking, it moved its arms to emphasize the conversation and to make it more pleasant. These movements were controlled by an RBN (Appendix C).). By the end of this activity, the students became familiar with the robot and therefore the environment was ready for the second activity.

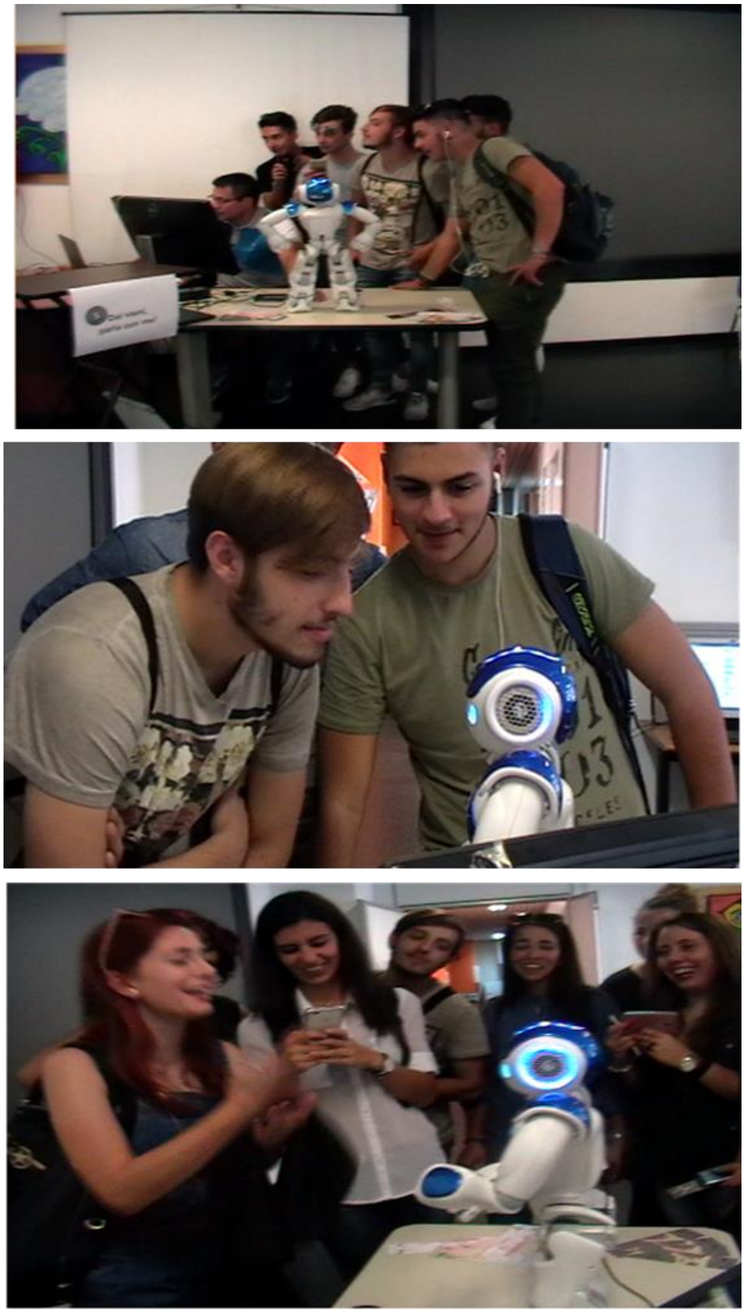


Figure 55. Quiz activity. The NAO Robot asked Italian general culture questions and assessed the answers from the students (video link).

The second task "**Come ti senti oggi?**" (How do you feel today?), was related to emotions interpretation. In this activity, the robot reacted to the three facial expressions described previously (happiness, sadness and surprise) which were imitated by the students. The NAO Robot was able to recognize and express itself easily to the happiness and surprise facial expressions; however, the students found difficulties to imitate the sad facial expression. therefore, it was difficult for the robot to recognize it. In this case, the students had to accentuate their facial expressions in order to get a reaction from the NAO Robot (Figure 56).

The last task for the robot was to recognize the hand gestures. In this activity "**Batte i cinque**" (Give me five) the robot had to imitate the movements and play the sound of some animals: duck, dog, cat, and horse in correspondence to the numbers 2, 3, 4 and 5 respectively; the numbers were represented by the fingers (Figure 57). This was the most difficult task to do because of robot movement and image noise (presence of colors similar to the skin on the surroundings). In order to assist this activity and to ensure the correct hand position so it can be detected; the images captured by the robot were projected on a monitor in which a green box indicated the best location. Additionally, a white backdrop was used as background to improve the hand detection.

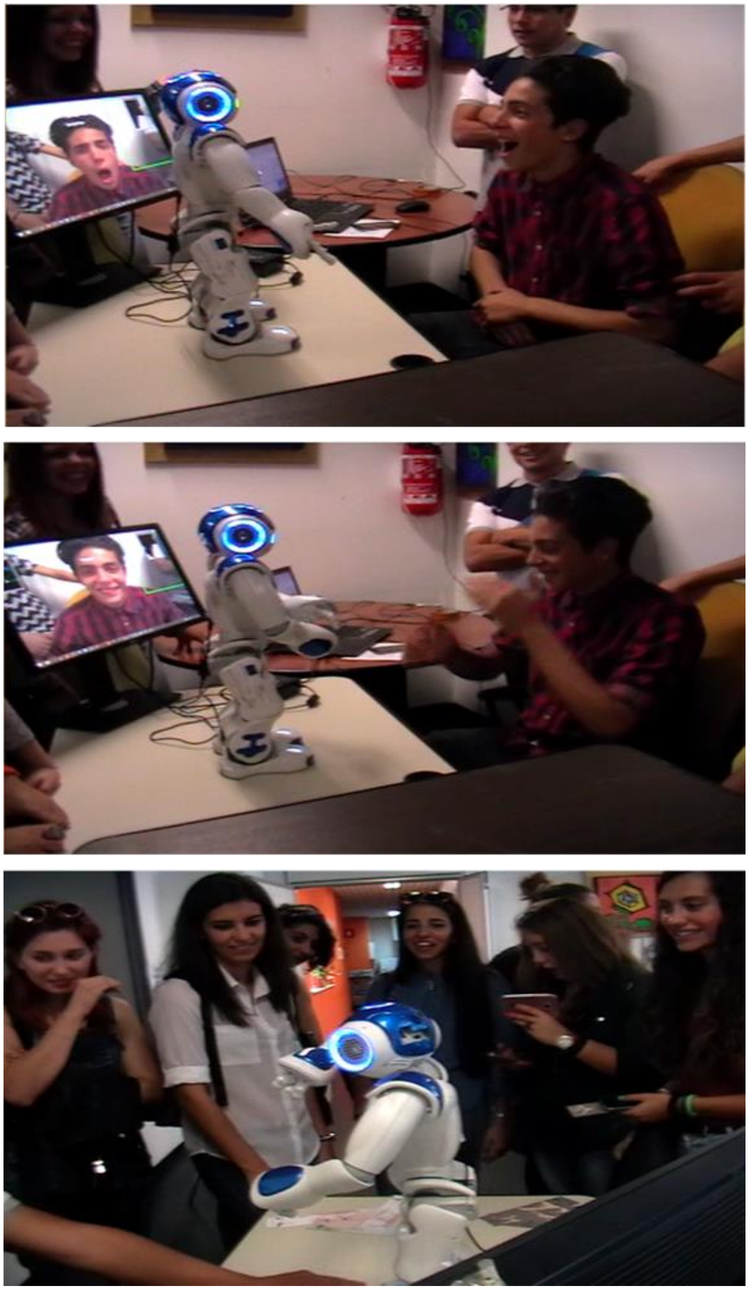


Figure 56. Activity: How do you feel today? The NAO Robot was able to recognize facial expressions and express emotions. It was easy for the NAO Robot to recognize happiness and surprise, but not sadness (video link).

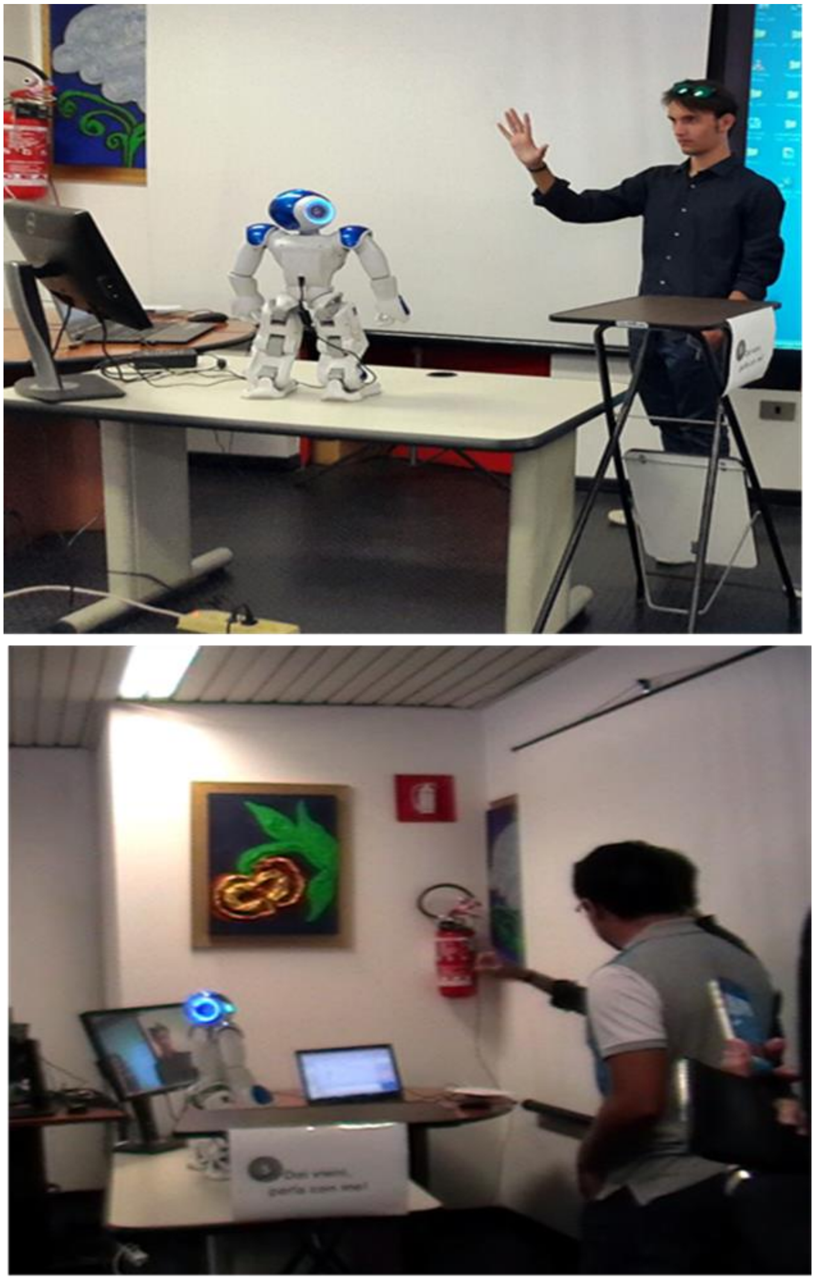


Figure 57. Activity: give me five!. The NAO Robot was able imitate the movements and play the sound of some animals: duck, dog, cat, and horse in correspondence to the numbers 2, 3, 4 and 5 respectively; the numbers were represented by the fingers.

Chapter 6

CONCLUSIONS AND FUTURE WORK

This project focused mainly on the development of a face recognition system, facial expressions and hand gestures for the NAO Robot and its integration into the Chroregraphe programming platform. This integration allowed to empower the NAO robot with essential HRI skills, making it a useful tool that it could be used by teachers in learning tasks or therapists in rehabilitation. Additionally, thankful to the telecontrol system, it can be easily operated from a remote site.

Human-robot interaction is essential allowing a robot be part of human life, either by assisting with basic tasks or by solving practical problems. Additionally, Vision-based system provides to the robot the capability to "understand" and express emotions to maintain an engaging relationship with people. It does not only provides better performance in assisting humans, but it also might improve robot's abilities to make decisions.

NAO Robot on-board computing resources are limited which making it difficult to implement sophisticated computer vision analysis algorithms which are needed for advanced interactive tasks. However, by implementing remote computing (Cloud computing), only basic tasks such as reading sensors or motion control are performed by the NAO Robot's CPU; other more complex tasks are carried out by a host wich has a better performance.

An interesting issue found during the experimental phase was the uncertainties during the facial expression recognition process by the robot; particularly it found difficulties to differentiate between neutral and sadness facial expressions. This, because some people might have naturally pronounced facial features related to some emotions or found it hard to mimic facial emotions (e.g. sadness); making it difficult to recognize or distinguish these two type of facial expressions not only by the machine but also by any human being. On the other hand, happiness and surprise were the facial features easier to simulate and therefore easier to be recognized by the robot.

PCA and Discriminant Analysis (DA) have been proved to be one of the most successful techniques in face recognition. They require a controlled environment in order to obtain a high recognition rate [87]; however, in the application of HRI, the surrounding environment continually varies which decreases recognition performance.

In future, is expected to increase facial expression recognition performance by carrying out an eyes & eyebrows geometrical analysis; in addition it will be increased the number of facial emotions that to be recognized and expressed by the NAO Robot.

References

[1] H. Hwang and D. Matsumoto, “Functions of Emotions,” in R. Biswas-Diener & E. Diener (Eds), Noba textbook series: Psychology. Champaign, 2016.

[2] IMotions, Facial expression analysis. 2016.

[3] P. Ekman and W. V Friesen, “1971Constants across cultures in the face and emotion,” vol. 17, no. 2, pp. 124–129, 1971.

[4] P. Ekman, “Are there basic emotions?,” Psychol. Rev., vol. 99, no. 3, pp. 550–553, 1992.

[5] M. Batty and M. J. Taylor, “Early processing of the six basic facial emotional expressions,” Cogn. Brain Res., vol. 17, no. 3, pp. 613–620, 2003.

[6] L. R. Goldberg, “Introduction to this Special Issue on Human-Robot Interaction.,” Multivariate Behav. Res., vol. 39, no. 2, p. 153, 2004.

[7] K. Kosuge and Y. Hirata, “Human-Robot Interaction,” Robot. Biomimetics, 2004. ROBIO 2004. IEEE Int. Conf. on., pp. 8–11, 2004.

[8] Aldebaran Robotics, “Nao software documentation.,” 2012. [Online]. Available: http://doc.aldebaran.com/1-14/index.html.

[9] Y. M. Shukla, A. Tamba, S. Pandey, and P. Sharma, “A Review and Scope of Humanoid Robotics,” no. March, pp. 28–29, 2014.

[10] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, “Vision-based corridor navigation for humanoid robots,” Proc. - IEEE Int. Conf. Robot. Autom., pp. 3190–3195, 2013.

[11] S. Hashimoto, S. Narita, H. Kasahara, K. Shirai, T. Kobayashi, a Takanishi, S. Sugano, J. Yamaguchi, H. Sawada, H. Takanobu, K. Shibuya, T. Morita, T. Kurata, N. Onoe, K. Ouchi, T. Noguchi, Y. Niwa, S. Nagayama, H. Tabayashi, I. Matsui, M. Obata, H. Matsuzaki, a Murasugi, S. Haruyama, T. Okada, Y. Hidaki, Y. Taguchi, K. Hoashi, E. Morikawa, Y. Iwano, D. Araki, J. Suzuki, M. Yokoyama, I. Dawa, D. Nishino, S. Inoue, T. Hirano, E. Soga, S. Gen, T. Yanada, K. Kato, S. Sakamoto, Y. Ishii, S. Matsuo, Y. Yamamoto, K. Sato, T. Hagiwara, T. Ueda, N. Honda, K. Hashimoto, T. Hanamoto, S. Kayaba, T. Kojima, H. Iwata, H. Kubodera, R. Matsuki, T. Nakajima, K. Nitto, D. Yamamoto, Y. Kamizaki, S. Nagaike, Y. Kunitake, and S., “Humanoid Robots in {Waseda} University--{Hadaly-2} and {WABIAN},” Auton. Robots, vol. 12, pp. 25–38, 2002.

[12] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, “ARMAR-III: An integrated humanoid platform for sensory-motor control,” Proc. 2006 6th IEEE-RAS Int. Conf. Humanoid Robot. HUMANOIDS, pp. 169–175, 2006.

[13] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, “The intelligent ASIMO: system overview and integration,” IEEE/RSJ Int. Conf. Intell. Robot. Syst., vol. 3, no. October, pp. 2478–2483, 2002.

[14] M. Fujita, “AIBO: Toward the Era of Digital Creatures,” Int. J. Rob. Res., vol. 20, no. 10, pp. 781–794, 2001.

[15] T. Kanda, H. Ishiguro, M. Imai, and T. Ono, “Development and evaluation of

interactive humanoid robots," Proc. IEEE, vol. 92, no. 11, pp. 1839–1850, 2004.

[16] S. Behnke and S. Behnke, "Humanoid Robots From Fiction to Reality?," KI-Zeitschrift, vol. 4, no. December, pp. 5–9, 2008.

[17] HONDA, "History of Honda's Robot Development," 2016. [Online]. Available: http://world.honda.com/ASIMO/history/.

[18] Waseda Humanoid Lab, "History of Humanoid Robot in Waseda University." [Online]. Available: http://www.humanoid.waseda.ac.jp/history.html.

[19] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. O. Lim, and A. Takanishi, "Development of a new humanoid robot WABIAN-2," Proc. - IEEE Int. Conf. Robot. Autom., vol. 2006, no. May, pp. 76–81, 2006.

[20] C. C. Wong, C. T. Cheng, K. H. Huang, Y. T. Yang, H. M. Chan, and C. S. Yin, "Mechanical design of small-size humanoid robot TWNHR-3," IECON Proc. (Industrial Electron. Conf., pp. 451–454, 2007.

[21] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," Neural Networks, vol. 23, no. 8–9, pp. 1125–1134, 2010.

[22] D. Vernon, G. Metta, and G. Sandini, "The iCub cognitive architecture: Interactive development in a humanoid robot," 2007 IEEE 6th Int. Conf. Dev. Learn. ICDL, pp. 122–127, 2007.

[23] G. Metta, L. Natale, F. Nori, and G. Sandini, "The iCub project: An open source platform for research in embodied cognition," Proc. IEEE Work. Adv. Robot. its Soc. Impacts, ARSO, pp. 24–26, 2011.

[24] V. K. Nath and S. E. Levinson, "Learning to Fire at Targets by an iCub Humanoid Robot," pp. 33–36, 2013.

[25] P. Kormushev, S. Calinon, R. Saegusa, and G. Metta, "Learning the skill of archery by a humanoid robot iCub," 2010 10th IEEE-RAS Int. Conf. Humanoid Robot. Humanoids 2010, no. 5, pp. 417–423, 2010.

[26] M. Frank, J. Leitner, M. Stollenga, G. Kaufmann, S. Harding, A. F&#246;rster, and Jurgen Schmidhuber, "The modular behavioral environment for humanoids and other robots (MoBeE)," ICINCO 2012 - Proc. 9th Int. Conf. Informatics Control. Autom. Robot., vol. 2, pp. 304–313, 2012.

[27] J. Leitner, S. Harding, M. Frank, A. Forster, and J. Schmidhuber, "Transferring spatial perception between robots operating in a shared workspace," IEEE Int. Conf. Intell. Robot. Syst., pp. 1507–1512, 2012.

[28] Park I.W., Kim J.Lee, and J. H. Oh, "Mechanical design of the humanoid robot platform HUBO," J. Adv. Robot., vol. 21, no. 11, pp. 1305–1322, 2007.

[29] I. W. Park, J. Y. Kim, J. Lee, and J. H. Oh, "Mechanical design of humanoid robot platform KHR-3 (KAIST humanoid robot - 3: HUBO)," Proc. 2005 5th IEEE-RAS Int. Conf. Humanoid Robot., vol. 2005, pp. 321–326, 2005.

[30] K. Kaneko, K. Harada, F. Kanehiro, G. Miyamori, and K. Akachi, "Humanoid robot HRP-3," Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 2471–2478, 2008.

[31] I. Park, J. Kim, J. Lee, M. Kim, and B. Cho, "Korea Advanced Institute of Science

and Technology (KAIST)," vol. 2, no. June, pp. 2–4, 2007.

[32] R. O. Flaherty, P. Vieira, M. Grey, P. Oh, A. Bobick, M. Egerstedt, and M. Stilman, "Kinematics and Inverse Kinematics for the Humanoid Robot HUBO2 +," pp. 2–7, 2013.

[33] I. Spectrum, "How South Korea's DRC-HUBO Robot Won the DARPA Robotics Challenge," 2015. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/humanoids/how-kaist-drc-hubo-won-darpa-robotics-challenge.

[34] M. Bibuli, M. Caccia, and L. Lapierre, "Path-following algorithms and experiments for an autonomous surface vehicle," IFAC Proc. Vol., vol. 7, no. PART 1, pp. 81–86, 2007.

[35] D. Agency Defense Advanced Research Projects, "DARPA Robotics Challenge," 2015. [Online]. Available: http://archive.darpa.mil/roboticschallenge/.

[36] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," 2009 IEEE Int. Conf. Robot. Autom., pp. 769–774, 2009.

[37] M. A. Miskam, S. Shamsuddin, M. R. A. Samat, H. Yussof, H. A. Ainudin, and A. R. Omar, "Humanoid robot NAO as a teaching tool of emotion recognition for children with autism using the Android app," 2014 Int. Symp. Micro-NanoMechatronics Hum. Sci. MHS 2014, pp. 2–6, 2015.

[38] E. Csala, G. Németh, and C. Zainkó, "Application of the nao humanoid robot in the treatment of bone marrowtransplanted children (demo)," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, pp. 2664–2666, 2013.

[39] A. Eguchi, "Robotics as a Learning Tool for Educational Transformation," Int. Work. Teach. Robot. Teach. with Robot. Int. Conf. Robot. Educ., pp. 27–34, 2014.

[40] D. R. Tobergte and S. Curtis, "Walking together hand in hand: Design and evaluation of autonomous robot system that a robot recognizes moving direction with a child's assistance of pulling its hand," J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2013.

[41] J. Lafaye, D. Gouaillier, P. Wieber, and I. Grenoble, "Linear Model Predictive Control of the locomotion of Pepper , a humanoid robot with omnidirectional wheels," 2014.

[42] F. Tanaka, K. Isshiki, F. Takahashi, M. Uekusa, R. Sei, and K. Hayashi, "Pepper learns together with children: Development of an educational application," IEEE-RAS Int. Conf. Humanoid Robot., vol. 2015-Decem, pp. 270–275, 2015.

[43] N. Kofinas, E. Orfanoudakis, and M. G. Lagoudakis, "Complete analytical inverse kinematics for NAO," Proc. 2013 13th Int. Conf. Auton. Robot Syst. Robot. 2013, 2013.

[44] S. Mukherjee, D. Paramkusam, and S. K. Dwivedy, "Inverse kinematics of a NAO humanoid robot using kinect to track and imitate human motion," Proc. 2015 Int. Conf. Robot. Autom. Control Embed. Syst. RACE 2015, no. February, 2015.

[45] D. Gouaillier and P. Blazevic, "A mechatronic platform, the Aldebaran Robotics humanoid robot," IECON Proc. (Industrial Electron. Conf., pp. 4049–4053, 2006.

[46] S. A. Nugroho, A. S. Prihatmanto, and A. S. Rohman, "Design and

implementation of kinematics model and trajectory planning for NAO humanoid robot in a tic-tac-toe board game," Proc. 2014 IEEE 4th Int. Conf. Syst. Eng. Technol. ICSET 2014, 2014.

[47] N. Kofinas, E. Orfanoudakis, and M. G. Lagoudakis, "Complete Analytical Forward and Inverse Kinematics for the NAO Humanoid Robot," J. Intell. Robot. Syst. Theory Appl., vol. 77, no. 2, pp. 251–264, 2014.

[48] J. Figat and W. Kasprzak, "NAO-mark vs QR-code recognition by Nao robot vision."

[49] E. Pot, J. Monceaux, R. Gelin, B. Maisonnier, and A. Robotics, "Choregraphe: A graphical tool for humanoid robot programming," Proc. - IEEE Int. Work. Robot Hum. Interact. Commun., pp. 46–51, 2009.

[50] E. Domingues, N. Lau, B. Pimentel, N. Shafii, L. P. Reis, and A. J. R. Neves, "Humanoid behaviors: From simulation to a real robot," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7026 LNAI, pp. 352–364, 2011.

[51] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "The NAO humanoid: a combination of performance and affordability," Cornell Univ. arXiv0807.3223 [cs.RO], pp. 1–10, 2008.

[52] C. Breazeal, "Emotion and sociable humanoid robots," Int. J. Hum. Comput. Stud., vol. 59, no. 1–2, pp. 119–155, 2003.

[53] A. R. Taheri, M. Alemi, A. Meghdari, H. R. PourEtemad, and S. L. Holderread, "Clinical Application of Humanoid Robots in Playing Imitation Games for Autistic Children in Iran," Procedia - Soc. Behav. Sci., vol. 176, no. July 2016, pp. 898–906, 2015.

[54] Y. C. Cho and J. W. Jeon, "Remote robot control system based on dtmf of mobile phone," IEEE Int. Conf. Ind. Informatics, pp. 1441–1446, 2008.

[55] M. Ogawa, K. Honda, Y. Sato, S. Kudoh, T. Oishi, and K. Ikeuchi, "Motion generation of the humanoid robot for teleoperation by task," 2015.

[56] L. Ismail, S. Shamsuddin, H. Hashim, S. Bahari, and A. Jaafar, "Face Detection Technique of Humanoid Robot NAO for Application in Robotic Assistive Therapy," pp. 517–521, 2011.

[57] S. Anila and N. Devarajan, "Global and Local Classifiers for Face Recognition," European Journal of Scientific Research, vol. 57, no. 4. pp. 556–566, 2011.

[58] L. Zhang, M. Jiang, D. Farid, and M. A. Hossain, "Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot," Expert Syst. Appl., vol. 40, no. 13, pp. 5160–5168, 2013.

[59] _ Open Source Computer VISION, "OPENCV3," 2015. [Online]. Available: http://docs.opencv.org/3.1.0/index.html#gsc.tab=0.

[60] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Comput. Vis. Pattern Recognit., vol. 1, pp. I–511–I–518, 2001.

[61] J. Howse, OpenCV Computer Vision with Python. 2013.

[62] P. Wagner, "Face Recognition with OpenCV2," Order A J. Theory Ordered Sets Its Appl., pp. 1–26, 2012.

[63] S. Raychaudhuri, J. M. Stuart, and R. B. Altman, “Principal components analysis to summarize microarray experiments: application to sporulation time series.,” Pac. Symp. Biocomput., pp. 455–66, 2000.

[64] H. C. Kim, D. Kim, and S. Yang Bang, “Face recognition using the mixture-of-eigenfaces method,” Pattern Recognit. Lett., vol. 23, no. 13, pp. 1549–1558, 2002.

[65] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces Vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” vol. 19, no. 7, pp. 711–720, 1997.

[66] K. Mistry, L. Zhang, S. C. Neoh, M. Jiang, A. Hossain, and B. Lafon, “Intelligent Appearance and shape based facial emotion recognition for a humanoid robot,” Ski. 2014 - 8th Int. Conf. Software, Knowledge, Inf. Manag. Appl., 2014.

[67] Free-Management-eBooks, “Effective Communication in the Workplace - Attitudes to Communications,” 2015. [Online]. Available: http://www.free-management-ebooks.com/faqcm/effective-05.htm.

[68] E. Antonakos, A. Roussos, and S. Zafeiriou, “A survey on mouth modeling and analysis for Sign Language recognition,” 2015 11th IEEE Int. Conf. Work. Autom. Face Gesture Recognition, FG 2015, no. August 2016, 2015.

[69] P. Ekman and E. L. Rosenberg, What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS). 2012.

[70] A. Maalej, B. Ben Amor, M. Daoudi, A. Srivastava, S. Berretti, B. Ben Amor, M. Daoudi, A. Srivastava, and S. Berretti, “Local 3D Shape Analysis for Facial Expression Recognition,” 2010 20th Int. Conf. Pattern Recognit., pp. 4129–4132, 2010.

[71] R. Zaier, THE FUTURE OF HUMANOID ROBOTS – RESEARCH AND Edited by Riadh Zaier. 2011.

[72] J. Wang, L. Yin, X. Wei, and Y. Sun, “3D facial expression recognition based on primitive surface feature distribution,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, pp. 1399–1406, 2006.

[73] J. Y. Baudouin and G. Tiberghien, “Symmetry, averageness, and feature size in the facial attractiveness of women,” Acta Psychol. (Amst)., vol. 117, no. 3, pp. 295–312, 2004.

[74] I. S. Penton-Voak, B. C. Jones, a C. Little, S. Baker, B. Tiddeman, D. M. Burt, and D. I. Perrett, “Symmetry, sexual dimorphism in facial proportions and male facial attractiveness.,” Proc. Biol. Sci., vol. 268, no. 1476, pp. 1617–23, 2001.

[75] G. Rhodes, F. Proffitt, J. M. Grady, and A. Sumich, “Facial symmetry and the perception of beauty,” Psychon. Bull. Rev., vol. 5, no. 4, pp. 659–669, 1998.

[76] Humintell, “The Seven Basic Emotions: Do you know them?,” 2016. [Online]. Available: http://www.humintell.com/2010/06/the-seven-basic-emotions-do-you-know-them/.

[77] M. Alsheakhali, A. Skaik, M. Aldahdouh, and M. Alhelou, “Hand Gesture Recognition System,” Comput. Eng., p. 5, 2011.

[78] J. M. Iverson and S. Goldin-Meadow, “Why people gesture when they speak.,” Nature, vol. 396, no. 6708, p. 228, 1998.

[79] J. Han, N. Campbell, K. Jokinen, and G. Wilcock, “Investigating the use of Non-

verbal Cues in Human-Robot Interaction with a Nao robot," 3rd IEEE Int. Conf. Cogn. Infocommunications, CogInfoCom 2012 - Proc., pp. 679–683, 2012.

[80] K. Jokinen and G. Wilcock, "Multimodal open-domain conversations with the Nao robot," Nat. Interact. with Robot. Knowbots …, pp. 1–12, 2014.

[81] Y. Wu and T. S. Huang, "Vision-Based Gesture Recognition: A Review," Gesture-Based Commun. Human-Computer Interact. Int. GestureWorkshop, GW'99 Gif-sur-Yvette, Fr. March 17-19, 1999 Proc., pp. 103–115, 1999.

[82] G. Bradski and A. Kaehler, Learning OpenCV. 2008.

[83] A. De Beir, H. Cao, P. Gomez, G. Van De Perre, and B. Vanderborght, "Enhancing Nao Expression of Emotions Using Pluggable Eyebrows."

[84] M. Häring, "HUMAN ROBOT INTERACTION."

[85] I. Cohen, R. Looije, and M. a. Neerincx, "Child's recognition of emotions in robot's face and body," Proc. 6th Int. Conf. Human-robot Interact. - HRI '11, p. 123, 2011.

[86] M. A. Miskam, S. Shamsuddin, H. Yussof, A. R. Omar, and M. Z. Muda, "Programming platform for NAO robot in cognitive interaction applications," 2014 IEEE Int. Symp. Robot. Manuf. Autom. IEEE-ROMA2014, pp. 141–146, 2015.

[87] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," Proceedings. 1991 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 586–591, 1991.

[88] filezilla-project, "FileZilla Client Tutorial," 2014. [Online]. Available: https://wiki.filezilla-project.org/FileZilla_Client_Tutorial_(en).

[89] Wolfram, "Wolfram Language & System Documentation Center," 2016. [Online]. Available: http://reference.wolfram.com/language/guide/MachineLearning.html.

[90] M. Hopfensitz, C. Müssel, M. Maucher, and H. A. Kestler, "Attractors in Boolean networks: A tutorial," Comput. Stat., vol. 28, no. 1, pp. 19–36, 2013.

[91] B. Drossel, "Number of attractors in random Boolean networks," Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys., vol. 72, no. 1, 2005.

[92] S. a Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets.," J. Theor. Biol., vol. 22, no. 3, pp. 437–467, 1969.

[93] C. Gershenson, "Introduction to Random Boolean Networks," Proc. Work. Tutorials Ninth Int. Conf. Simul. Synth. Living Syst. (ALife IX), Boston, USA, pp. 160–173, 2004.

[94] K. Inoue, "Logic Programming for Boolean Networks," in Twenty-Second International Joint Conference on Artificial Intelligence, pp. 924–930.

[95] A. Roli, M. Manfroni, C. Pinciroli, and M. Birattari, "On the design of Boolean network robots," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6624 LNCS, no. PART 1, pp. 43–52, 2011.

[96] L. Cammoun, X. Gigandet, D. Meskaldji, J. P. Thiran, O. Sporns, K. Q. Do, P. Maeder, R. Meuli, and P. Hagmann, "Mapping the human connectome at multiple scales with diffusion spectrum MRI," J. Neurosci. Methods, vol. 203, no. 2, pp. 386–397, 2012.

[97] P. Hagmann, M. Kurant, X. Gigandet, P. Thiran, V. J. Wedeen, R. Meuli, and J. P. Thiran, “Mapping human whole-brain structural networks with diffusion MRI,” PLoS One, vol. 2, no. 7, 2007.

[98] M. B. Cuadra, C. Pollo*, A. Bardera, O. Cuisenaire, J.-G. . Villemure, and J.-P. Thiran, “ATLAS-BASED SEGMENTATION OF PATHOLOGICAL BRAIN MR IMAGES,” pp. 0–3, 2003.

APPENDICES

Appendix A

Running Python script on the robot

The NAO Robot needs to run some applications automatically at NAOqi startup. NAO's Choregraphe allows easily to install within the robot the applications using the "**Package and install**" button (Figure 58, box 1), the applications can be configured to run automatically using the "**Default**" button (Figure 58, box 2). However, to install and run a pure Python script, it requires an approach slightly different.

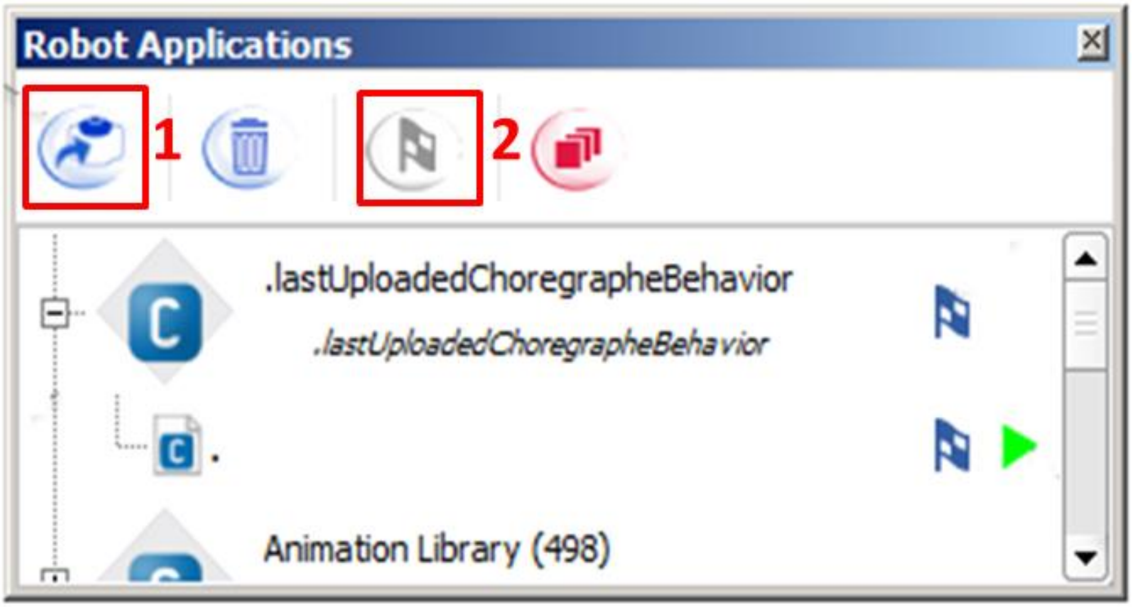
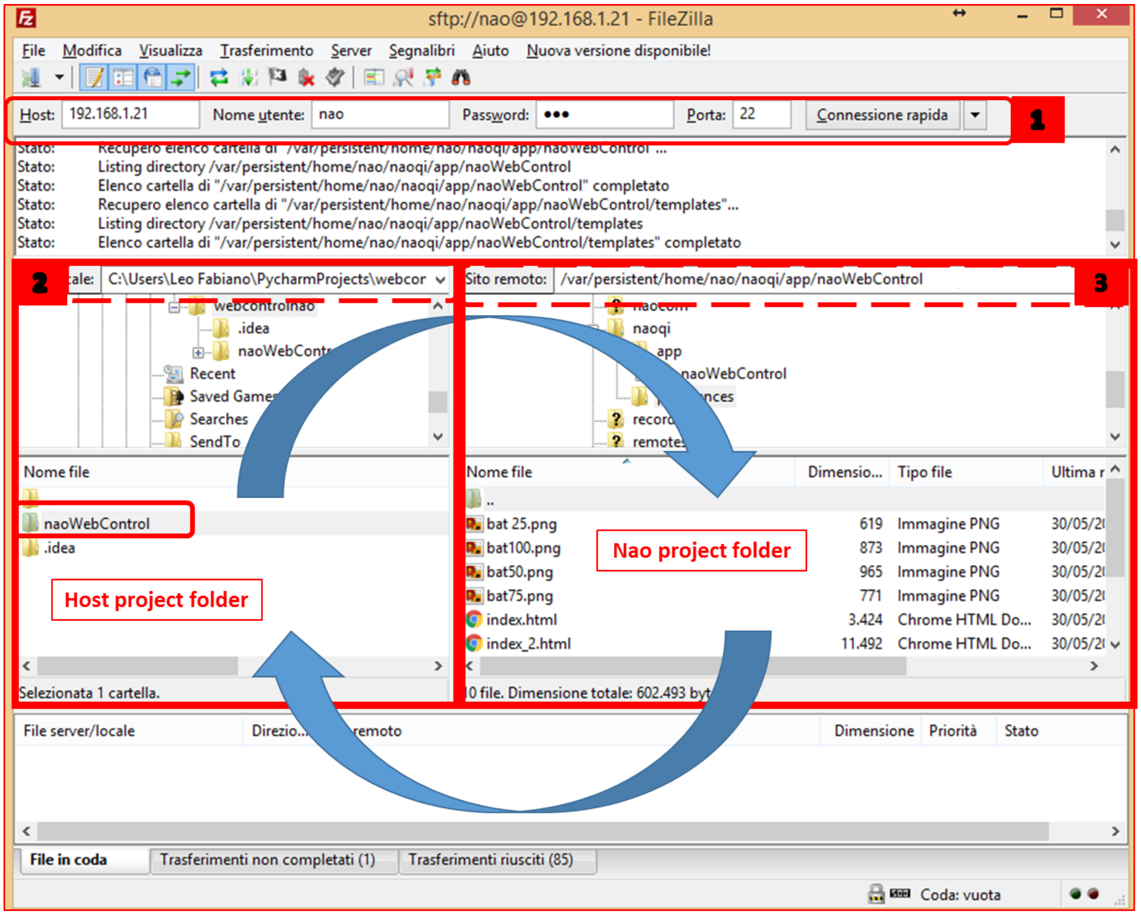

Robot Applications
1
2
.lastUploadedChoregrapheBehavior
.lastUploadedChoregrapheBehavior
Animation Library (498)

Figure 58. NAO Robot applications window. (1) Package and install button and (2) Default button.

For example, in order to launch automatically the **WWW Tele-control** application within NAO Robot; all files have to be transferred to the robot via FTP (File Transfer Protocol) program e.g. Filezilla (Figure 59). On the quickconnect bar (Figure 59, Box 1), the follow information should be entered: the IP address of the NAO Robot into **HOST:** field, the username "nao" into the **Username:** field as well as the password "nao" into the **Password:** field and port 22 into **Port:** field.


sftp://nao@192.168.1.21 - FileZilla

Host: 192.168.1.21 Nome utente: nao Password: ••• Porta: 22 Connessione rapida

```
Stato:     Recupero elenco cartella di "/var/persistent/home/nao/naoqi/app/naoWebControl"...
Stato:     Listing directory /var/persistent/home/nao/naoqi/app/naoWebControl
Stato:     Elenco cartella di "/var/persistent/home/nao/naoqi/app/naoWebControl" completato
Stato:     Recupero elenco cartella di "/var/persistent/home/nao/naoqi/app/naoWebControl/templates"...
Stato:     Listing directory /var/persistent/home/nao/naoqi/app/naoWebControl/templates
Stato:     Elenco cartella di "/var/persistent/home/nao/naoqi/app/naoWebControl/templates" completato
```

...ale: C:\Users\Leo Fabiano\PycharmProjects\webcor

Sito remoto: /var/persistent/home/nao/naoqi/app/naoWebControl

Nome file
..
naoWebControl
.idea

Selezionata 1 cartella.

Nome file	Dimensio...	Tipo file	Ultima r
..			
bat 25.png	619	Immagine PNG	30/05/2(
bat100.png	873	Immagine PNG	30/05/2(
bat50.png	965	Immagine PNG	30/05/2(
bat75.png	771	Immagine PNG	30/05/2(
index.html	3.424	Chrome HTML Do...	30/05/2(
index_2.html	11.492	Chrome HTML Do...	30/05/2(

0 file. Dimensione totale: 602.493 byt

File in coda | Trasferimenti non completati (1) | Trasferimenti riusciti (85)

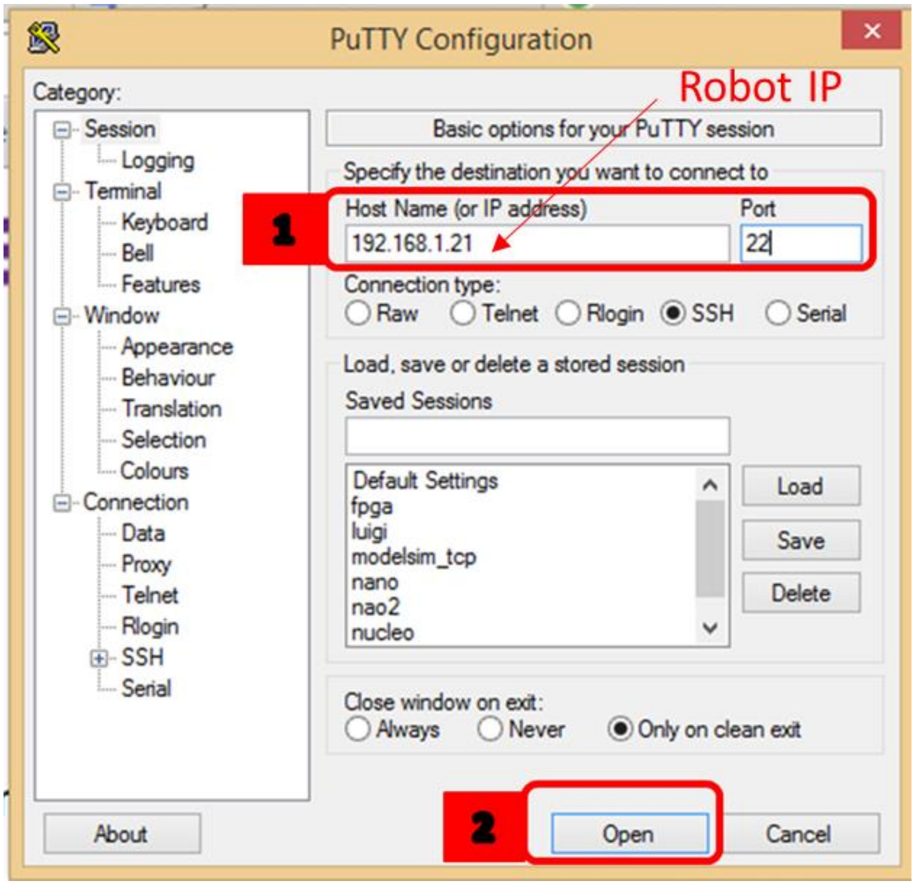
Coda: vuota

Figure 59. FTP file transfer. (1). Quick connect bar, (2) Local pane and (3) Remote pane.

Into the Navigating layout are located all files. The left column (Figure 59, Box 2) displays files and directories locally. The right column (Figure 59, Box 3) displays files and directories into NAO Robot. Both columns have on top a structure as directory and a detailed list of the contents of the selected directory at bottom. It is easily browse between this two trees and lists by clicking around like in any other file manager [88]. Finally, drag the folder with the project from left to right; this action can be done in both directions.

The next step is to run the scripts that control all functions through a SSH software like PUTTY (Figure 60). To establish communication login credentials must be provided, the user name and default password are "nao". In addition, you must log in as root using the command "su" and the root password "root".

Then, navigate to python scripts directory by using the command "cd /absolute/path/to/reach/the/source/folder". At this point, the command "python name_of _script.py" is used to run the script. It will be necessary to repeat the ssh session for video, WebSocket and HTTP servers (Figure 61). Finally, to set scripts as default run, the **“**/home/nao/naoqi/preferences/autoload.ini**”** file must be edited using the command “**nano**” and adding the Python scripts paths (Figure 62).



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.21

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

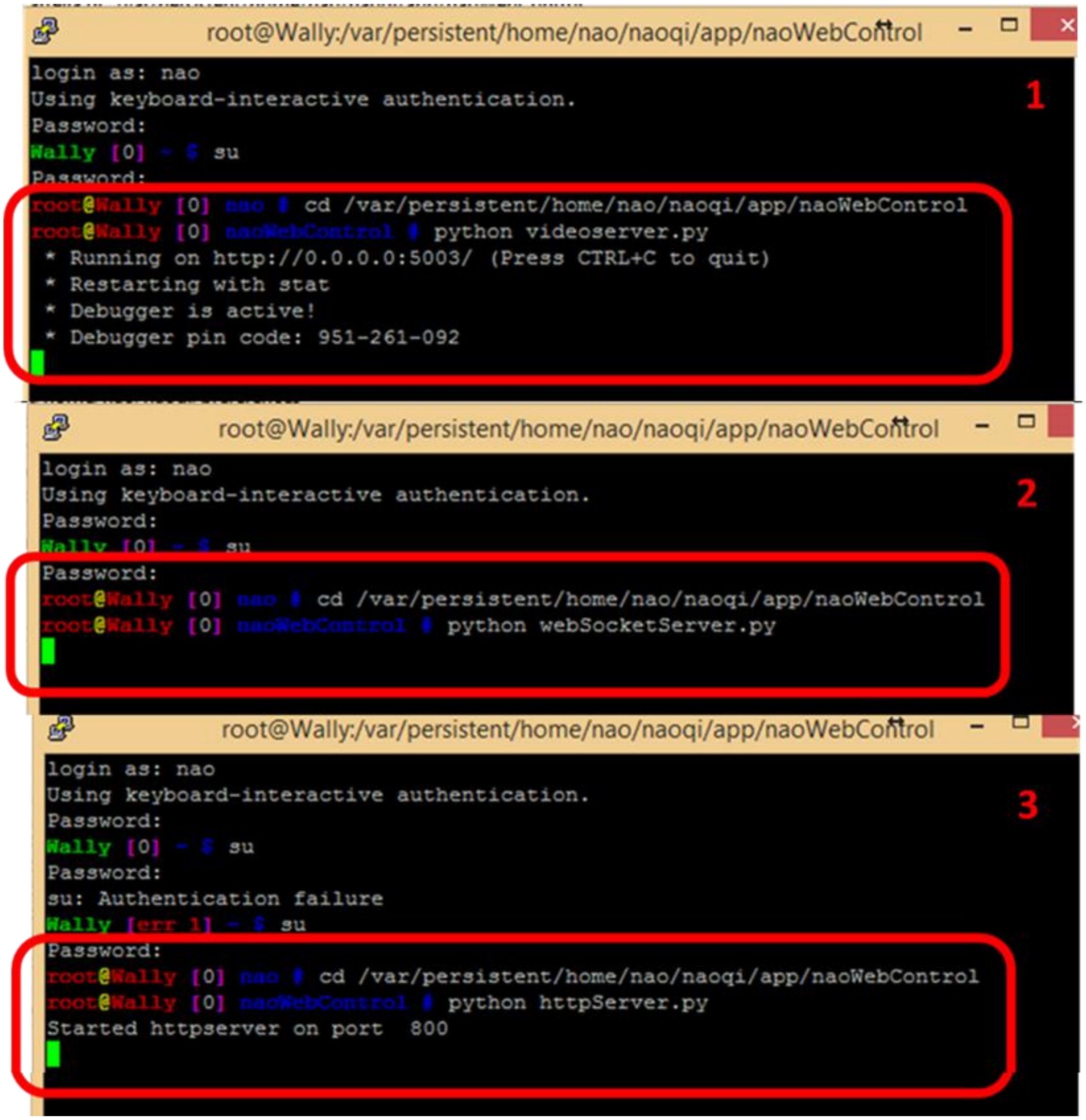
Load, save or delete a stored session

Saved Sessions

Default Settings
fpga
luigi
modelsim_tcp
nano
nao2
nucleo

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 60. SSH communication. (1) NAO Robot's IP address and port. (2) Connect button.

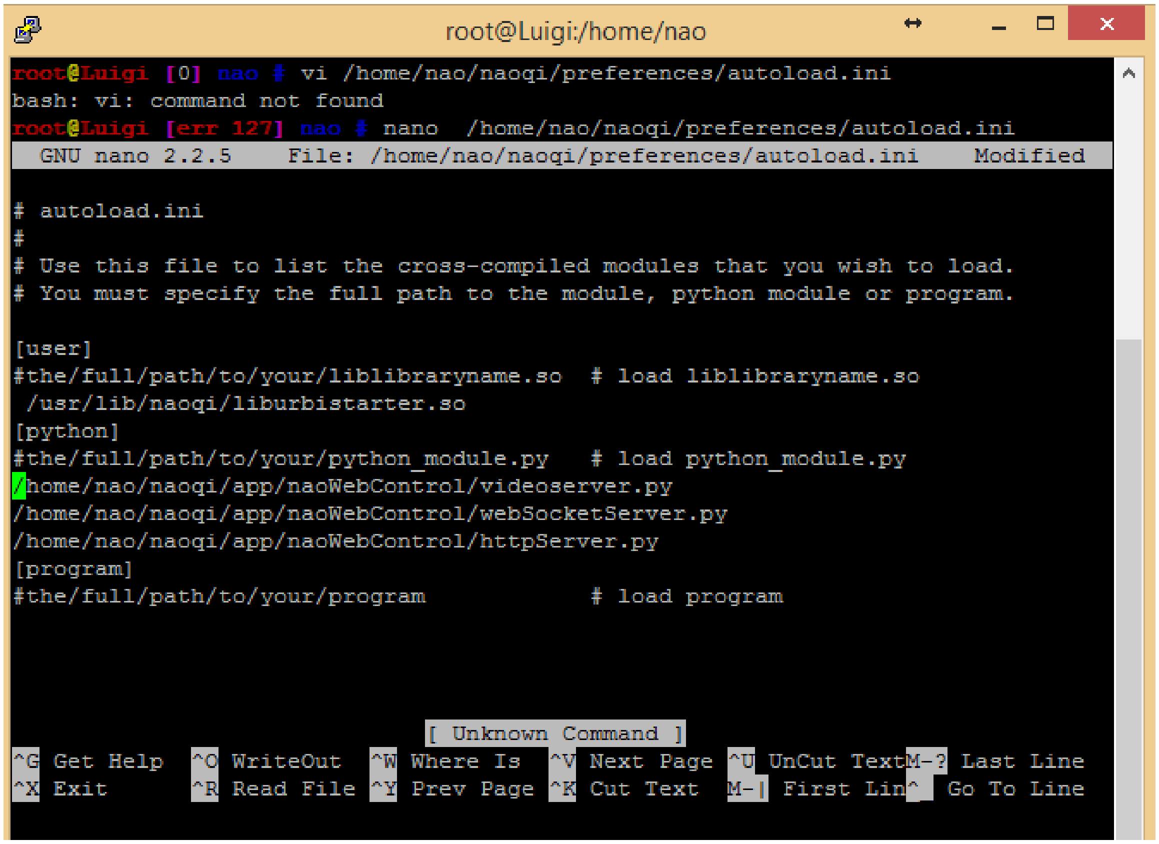

root@Wally:/var/persistent/home/nao/naoqi/app/naoWebControl
1
root@Wally:/var/persistent/home/nao/naoqi/app/naoWebControl
2
root@Wally:/var/persistent/home/nao/naoqi/app/naoWebControl
3

```
login as: nao
Using keyboard-interactive authentication.
Password:
Wally [0] ~ $ su
Password:
root@Wally [0] nao # cd /var/persistent/home/nao/naoqi/app/naoWebControl
root@Wally [0] naoWebControl # python videoserver.py
 * Running on http://0.0.0.0:5003/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger pin code: 951-261-092
```

```
login as: nao
Using keyboard-interactive authentication.
Password:
Wally [0] ~ $ su
Password:
root@Wally [0] nao # cd /var/persistent/home/nao/naoqi/app/naoWebControl
root@Wally [0] naoWebControl # python webSocketServer.py
```

```
login as: nao
Using keyboard-interactive authentication.
Password:
Wally [0] ~ $ su
Password:
su: Authentication failure
Wally [err 1] ~ $ su
Password:
root@Wally [0] nao # cd /var/persistent/home/nao/naoqi/app/naoWebControl
root@Wally [0] naoWebControl # python httpServer.py
Started httpserver on port  800
```

Figure 61. Execution of the Python script using ssh. (1) Video server. (2) Web Socket Server and (3) HTTP server.



```
root@Luigi [0] nao # vi /home/nao/naoqi/preferences/autoload.ini
bash: vi: command not found
root@Luigi [err 127] nao # nano  /home/nao/naoqi/preferences/autoload.ini
  GNU nano 2.2.5     File: /home/nao/naoqi/preferences/autoload.ini     Modified

# autoload.ini
#
# Use this file to list the cross-compiled modules that you wish to load.
# You must specify the full path to the module, python module or program.

[user]
#the/full/path/to/your/liblibraryname.so  # load liblibraryname.so
 /usr/lib/naoqi/liburbistarter.so
[python]
#the/full/path/to/your/python_module.py   # load python_module.py
/home/nao/naoqi/app/naoWebControl/videoserver.py
/home/nao/naoqi/app/naoWebControl/webSocketServer.py
/home/nao/naoqi/app/naoWebControl/httpServer.py
[program]
#the/full/path/to/your/program            # load program




                              [ Unknown Command ]
^G Get Help   ^O WriteOut   ^W Where Is   ^V Next Page  ^U UnCut TextM-? Last Line
^X Exit       ^R Read File  ^Y Prev Page  ^K Cut Text   M-| First Lin^  Go To Line
```

Figure 62. Edit "autoload.ini" file to running automatically Python sripts at NAOqi startup. Use "nano" command and "CTL+X" then "S" to save.

Appendix B

Mathematica based emotion classifier

The Wolfram Language includes a wide range of integrated machine learning capabilities, it provides highly automated functions for Prediction and Classification [89]. These functions allow different data types manipulation including numerical, categorical time series, textual and images.

A machine for classifying facial emotions (sad, happy, neutral) was implemented. Mathematica built in functions allow to create a classifier machine so easy and fast. Practically, the classifier machine keeps the follow sequence (Figure 29): first, image dataset is loaded, before training, face detection is performed, only if face is detected, the image is included in data set. Then, the new data is used in the training stage. Now, a new set of images is used to test the classifier, before test stage, face detection is performed again.

From 429 images used for training (Dataset) and the 281 images used to for data test (TestData), after face detection stage, the number of images of Dataset was reduced to 234 (Figure 63) and the TestData to 158 (Figure 64), a reducttion of about 54 % of the faces. With this new set of data, the rate of emotions recognition was about 90%; happiness had a recognition succes rate of 92%, sadness of 85% and neutral expression of 90%.

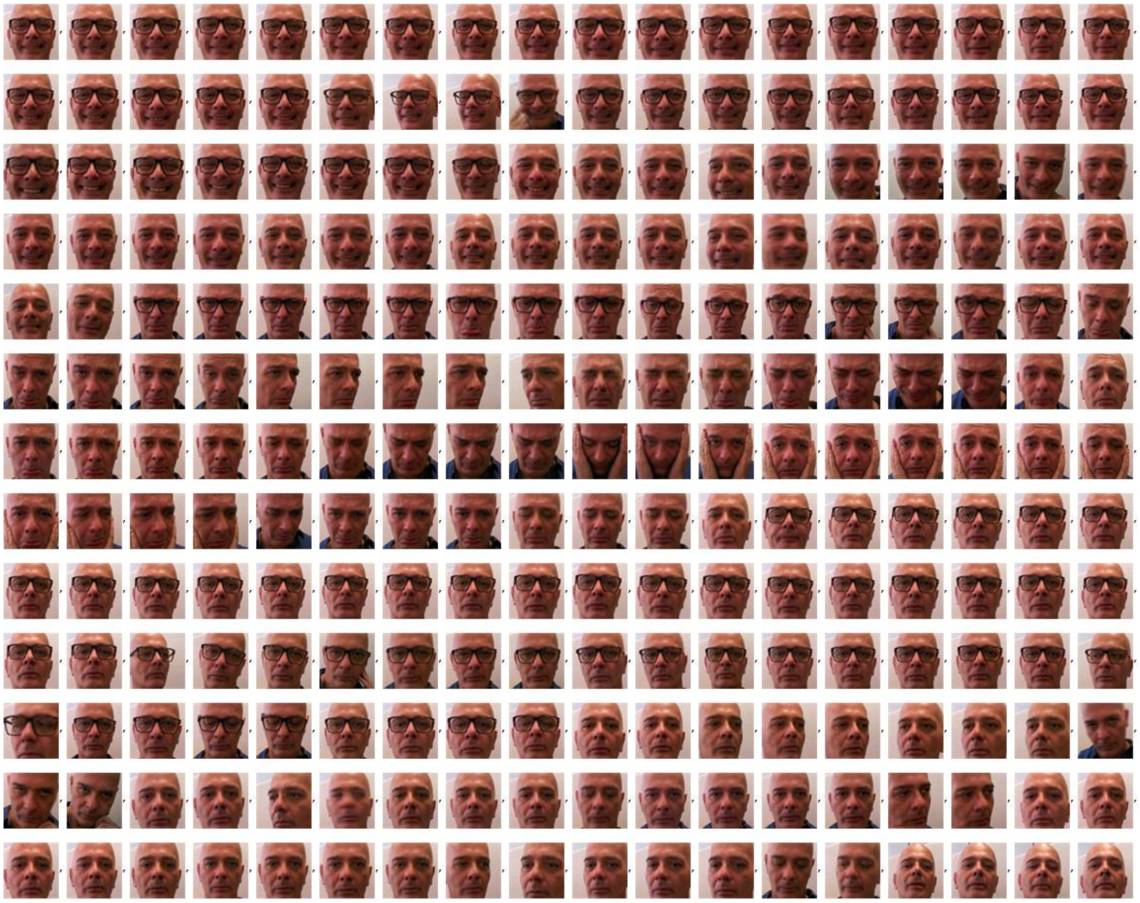


Figure 63. Mathematica emotions Data set images used to model the emotions classifier.

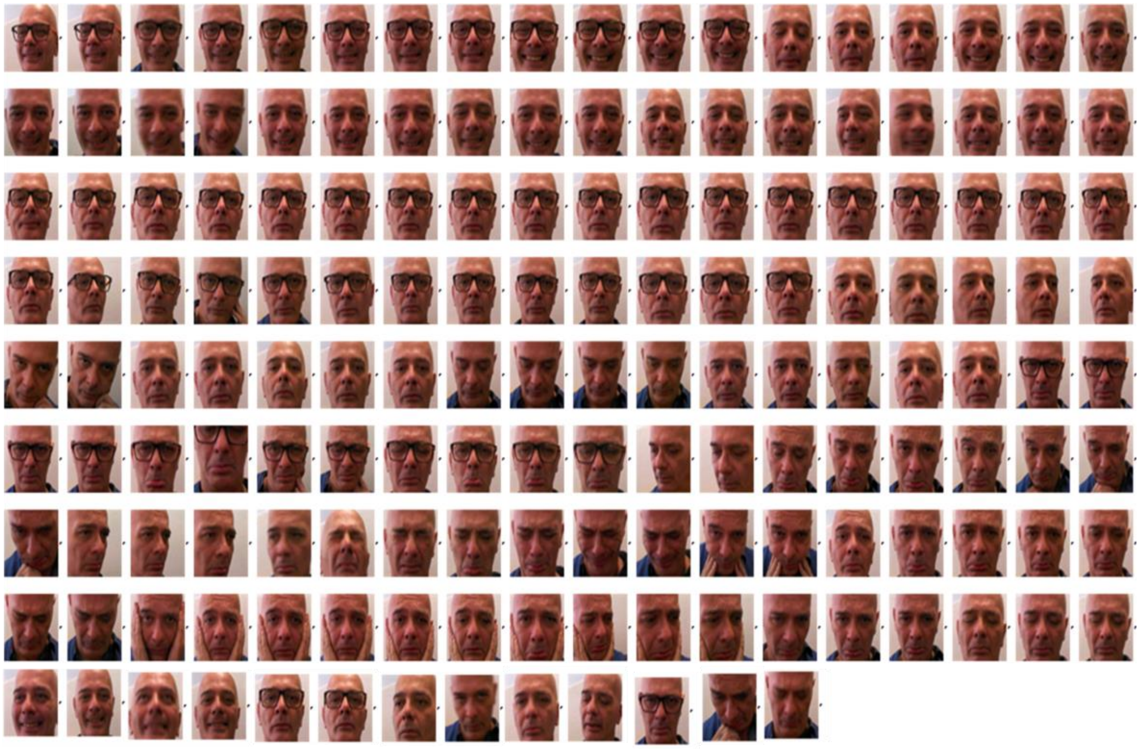


Figure 64. Mathematica emotions Test Data images used to test the emotions classifier.

Appendix C

Random pattern generator using Boolean networks

Boolean networks are qualitative descriptors of gene regulatory and dynamic models [90]. **Random Boolean Networks** (RBNs) are the simplest and most widely studied model [91], were introduced by Kauffman in 1969 [92] as a model for gene regulation. They are also known as N−K models, or Kauffman networks [93].

A Boolean network is a pair N = (V, F), where V = {v1,...,vn} is a finite set of nodes (or genes) and F = {f1,...,fn} is a corresponding set of Boolean functions or regulation functions [92], one for each variable. These transition functions map a Boolean input (variables in V) to a Boolean value 0 or 1 [90]. A BN is usually represented by a graph with two types of edges (Figure 65, box 1), which are positive and negative, in which v_1 ——→ v_2 means that v_1^t positively takes part in the regulation function for v_2^{t+1} and v_1 —–| v_2 means that v_1^t negatively takes part in the regulation function for v_2^{t+1}[94].

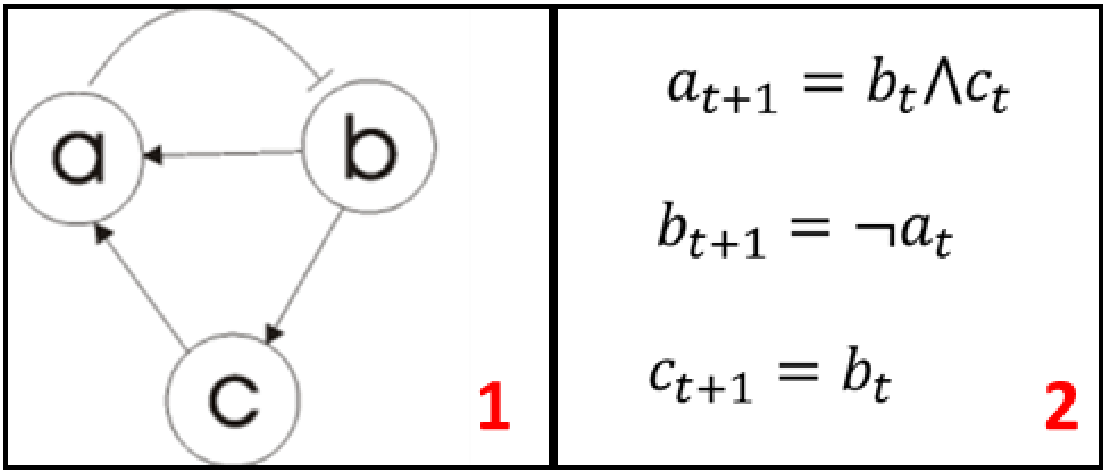


Figure 65. Boolean network. (1) Inputs diagram and (2) Boolean functions.

The nodes in classic RBNs are synchronous, the next state of nodes at time t+1 depend on the current state of nodes at time t so that all nodes “march in step”[93]. Random Boolean networks have been applied and used as models in many different areas, such as evolutionary theory, mathematics, sociology, neural networks, robotics, and music generation [93]. In this way, BNs can be used as robot program. The robot dynamics can be described in terms of trajectories (or motions) in a space of discrete states, allowing to design the robot program by directly exploiting the dynamical characteristics of BNs [95].

A complex network that represents the functional brain network was used to control NAO Robot’s arm motions. The network was mapped using Diffusion MRI techniques [96], [97]. It is composed by 83 nodes corresponding to large-scale cortical and subcortical regions (Figure 66) extracted from the Lausanne atlas [98].

The successive state of a node in the network depends on state of linked nodes, if the number of enabled nodes is in the range [a b], the node will be enabled (Figure 67, box 1), otherwise will be disabled (Figure 67, box 2).

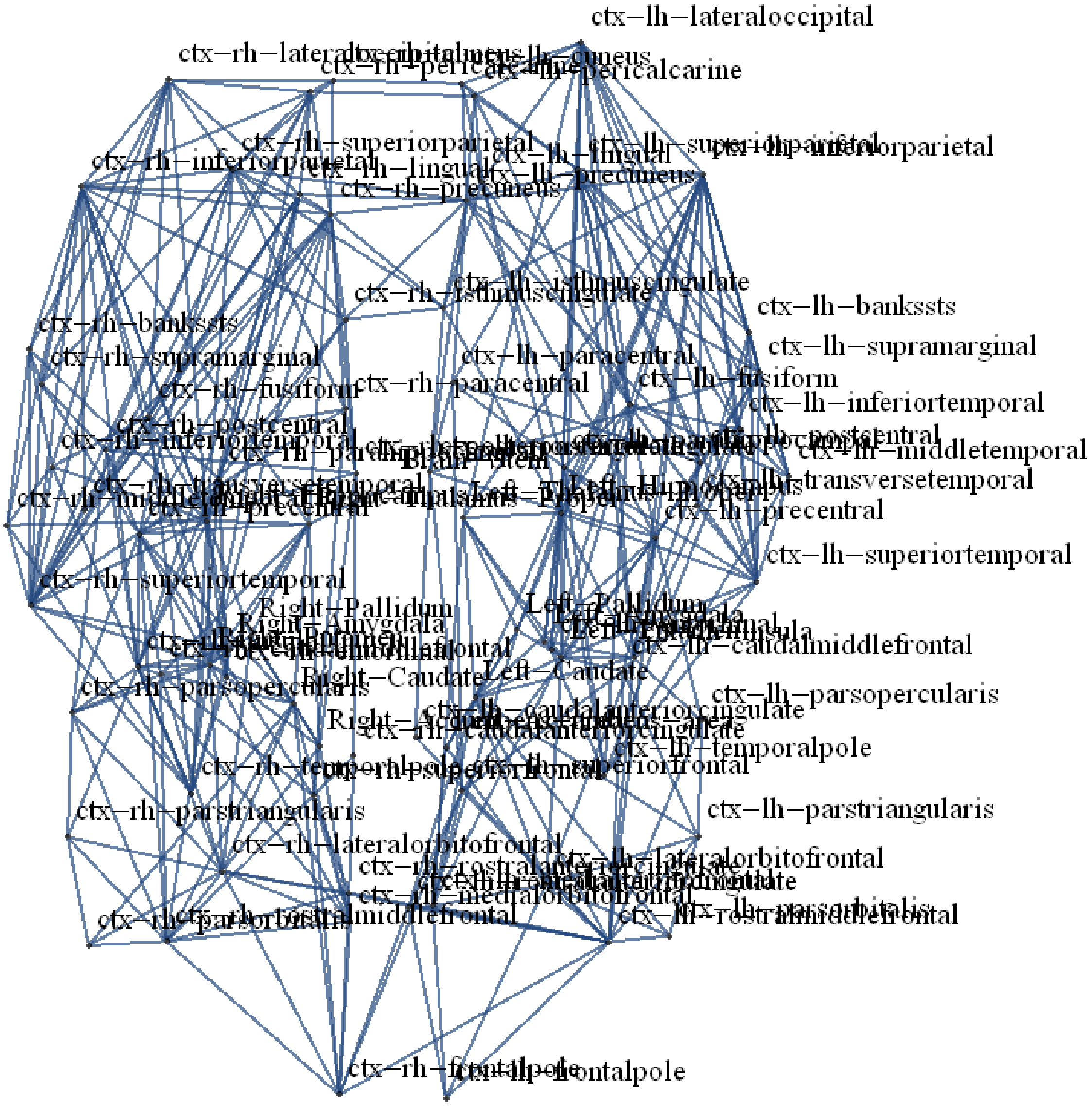


Figure 66. Structural connectivity of brain network with diffusion MRI. The *network of structural connectivity across the entire is composed of 83 nodes corresponding to large-scale cortical and subcortical regions.*

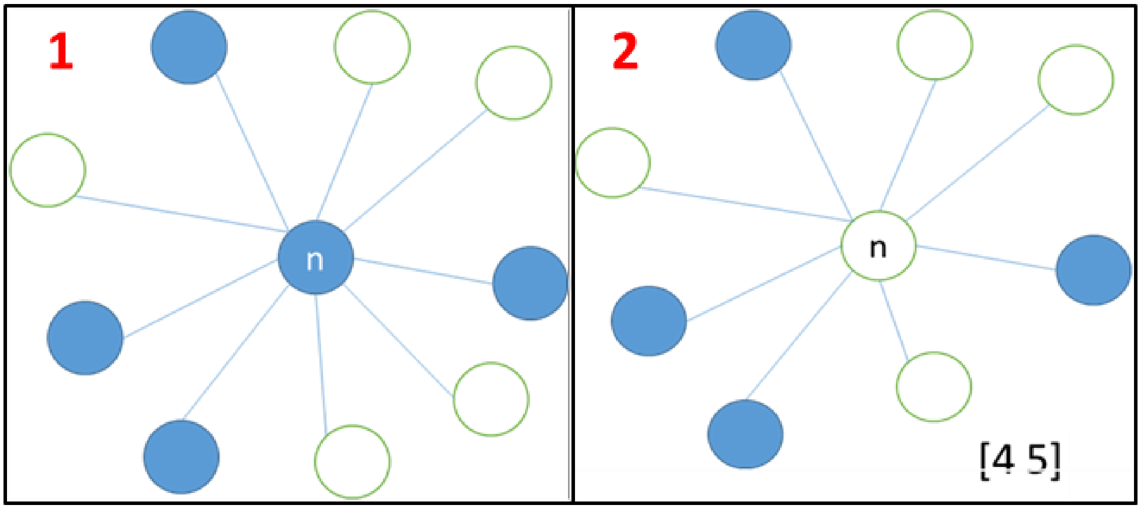


Figure 67. Network dynamics function. When the number of linked active nodes is between [4 5], the current node is activated (1), otherwise is deactivated (2).

The evolution matrix that represents the Dynamics of the RBN under random initial state (conditions) is used as pattern generator for the NAO Robot. Before extracting two vectors randomly, nodes that are all enabled or all disabled after 300 steps are rejected from the evolution matrix (Figure 68).

The evolution matrix is used as motion pattern generator, the pair of extracted vectors could be directly encoded e.g. as "00","01", "10" and "11" to drive robot arms to angles 0°, 50°, 40° and 60° respectively (Figure 69). The motion control is constrained by the NAOqi angle interpolation commad.

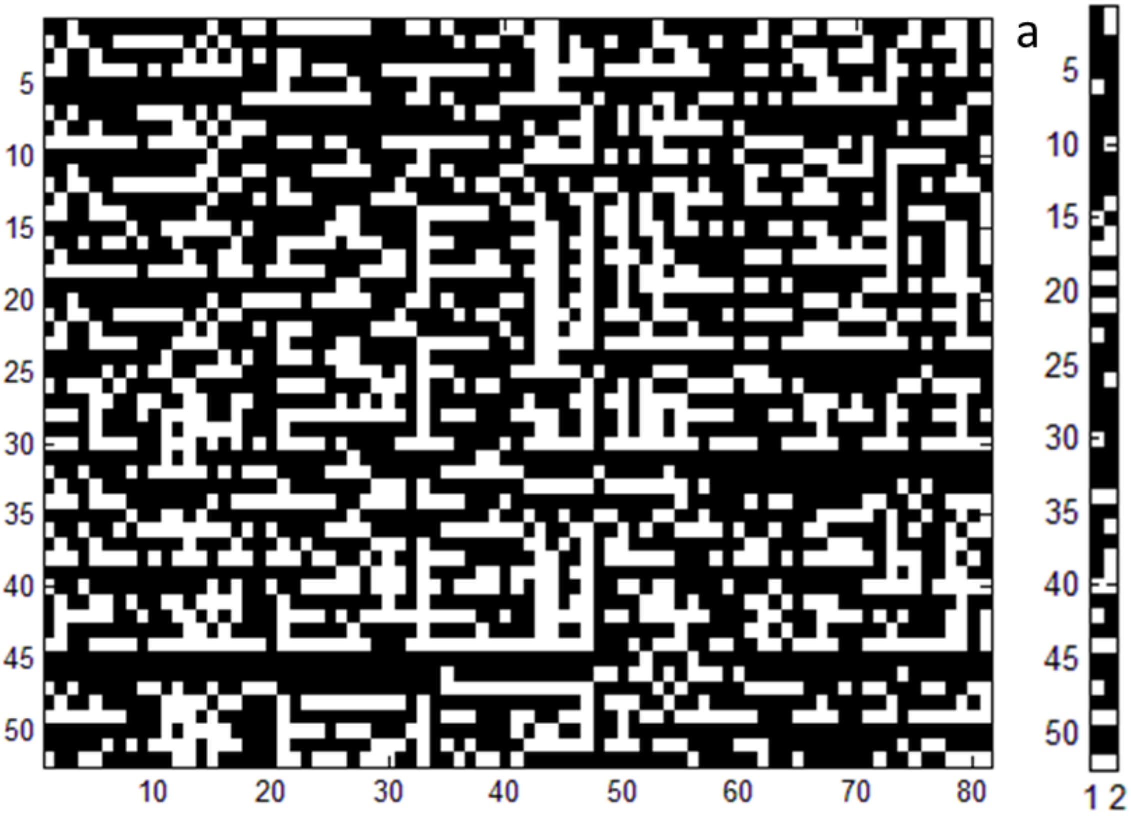


Figure 68. Dynamics of the brain network after 300 steps. (a) Evolution Matrix. (b) Extracted vectors.

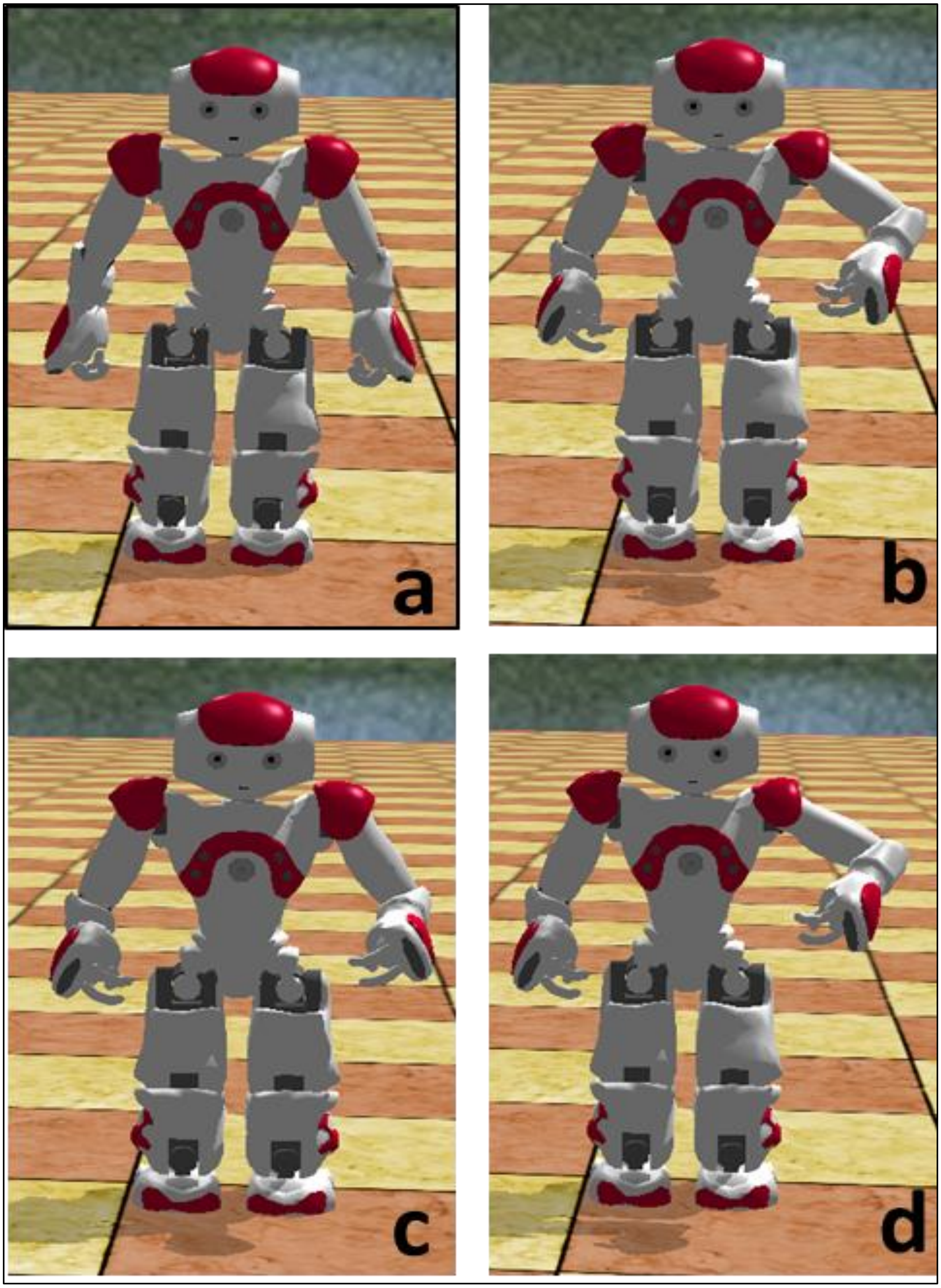


Figure 69. Arm LShoulderPitch angle control using a RBN. (a) 0°, (b) 50°, (c) 40° and (d) 60° (Video link).