UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

PH.D. THESIS

IN

INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

DEVELOPING AN AUTONOMOUS MOBILE ROBOTIC DEVICE FOR MONITORING AND ASSISTING OLDER PEOPLE

GIOVANNI ERCOLANO

TUTOR: PROF. SILVIA ROSSI

COORDINATOR: PROF. DANIELE RICCIO

XXXIII CICLO

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Abstract

A progressive increase of the elderly population in the world has required technological solutions capable of improving the life prospects of people suffering from senile dementias such as Alzheimer's. Socially Assistive Robotics (SAR) in the research field of elderly care is a solution that can ensure, through observation and monitoring of behaviors, their safety and improve their physical and cognitive health. A social robot can autonomously and tirelessly monitor a person daily by providing assistive tasks such as remembering to take medication and suggesting activities to keep the assisted active both physically and cognitively. However, many projects in this area have not considered the preferences, needs, personality, and cognitive profiles of older people. Moreover, other projects have developed specific robotic applications making it difficult to reuse and adapt them on other hardware devices and for other different functional contexts. This thesis presents the development of a scalable, modular, multi-tenant robotic application and its testing in real-world environments. This work is part of the UPA4SAR project "User-centered Profiling and Adaptation for Socially Assistive Robotics". The UPA4SAR project aimed to develop a low-cost robotic application for faster deployment among the elderly population. The architecture of the proposed robotic system is modular, robust, and scalable due to the development of functionality in microservices with event-based communication. To improve robot acceptance the functionalities, enjoyed through microservices, adapt the robot's behaviors based on the preferences and personality of the assisted person. A key part of the assistance is the monitoring of activities that are recognized through deep neural network models proposed in this work. The final experimentation of the project

carried out in the homes of elderly volunteers was performed with complete autonomy of the robotic system. Daily care plans customized to the person's needs and preferences were executed. These included notification tasks to remember when to take medication, tasks to check if basic nutrition activities were accomplished, entertainment and companionship tasks with games, videos, music for cognitive and physical stimulation of the patient.

Contents

Abstract i

1 Introduction 1

1.1 Problem Definition . . . 2
1.2 Our Solution . . . 4
1.3 Innovative Aspects . . . 6
1.4 Thesis' Contribution . . . 7
1.5 Structure of the Thesis . . . 8
1.6 Publications . . . 9

2 Related Work 11

2.1 Assistive Robot for Elderly Home Care . . . 11
2.2 Service-Oriented Architecture (SOA) . . . 19
2.3 ADL Classification Methods . . . 22
2.4 Summary . . . 26

3 Architecture 31

3.1 Functional Requirements . . . 31
3.2 The Proposed Architecture . . . 34
3.2.1 The proposed framework from planner perspective . . . 36
3.2.2 Daily Assistive Workflow Generator . . . 38
3.2.3 The software architecture . . . 40
3.2.4 IoT Devices . . . 42
3.2.5 Robot Server . . . 43

3.2.6 Implemented Services 45

4 Services **47**

4.1 Software Analysis Overview 47

4.1.1 Product outlook 48

4.1.2 Product Functionality 48

4.1.3 User characteristics 49

4.1.4 General Constraints 49

4.1.5 Assumptions and dependencies 49

4.1.6 Specific Requirements 49

4.2 Functional Requirements 50

4.3 Performance Requirements 57

4.3.1 Design Constraints 57

4.3.2 Software System Attributes - Security 57

4.4 Monitoring Services 57

4.5 Navigation Services 59

4.6 Interaction Services 60

4.7 Communication 61

4.8 Testing 63

4.8.1 Results 65

4.8.2 Discussion 66

5 ADL Recognition **67**

5.1 The Proposed Approach 69

5.2 Testing with dataset CAD-60 74

5.2.1 Dataset 76

5.2.2 Data Pre-processing 77

5.2.3 Model Settings 78

5.2.4 Implementation Details 78

5.2.5 Classification Results 79

5.2.6 Statistical Hypothesis Test 81

5.2.7 Window Size Results 82
5.2.8 Comparison with the SoA 82
5.2.9 Real Settings Configuration 85
5.3 Testing with our sampled dataset 86
5.3.1 Dataset 86
5.3.2 Data Pre-processing 87
5.3.3 Model Settings 88
5.3.4 Implementation Details 88
5.3.5 Classification Results 89
5.4 Additional Application about Robot Gesture Imitation performed by autistic children 89
5.4.1 Proposed Method 91
5.4.2 Settings 92
5.4.3 Comparison with classical ML methods 93
5.4.4 Results 94
5.4.5 Discussion 96

6 Field Experimentation **100**
6.1 Collected Information 101
6.2 Experimental Procedure 101
6.3 Scheduling and Execution of the Personalized Daily Assistive Plan . 103
6.4 Results on Acceptance 108
6.5 Patients Interviews 111

7 Discussion **113**

List of Figures

3.1 The proposed framework from the perspective of the planner 37
3.2 Example of an abstract workflow generated for the *Wake-Up* activity 39
3.3 The software architecture 41
3.4 Sanbot Elf robot and its hardware components 43
3.5 Control Interface for testing the System Components 44

4.1 A diagram with the principal IoT components that run the different services. 62
4.2 A sequence diagram example of the launch of a workflow. 64

5.1 A diagram of one LSTM 69
5.2 An example of a CNN: LeNet introduced by Yan LeCun [45] 70
5.3 An abstract diagram of the proposed three-dimensional matrix for human pose representation 71
5.4 Combination of a CNN for automatic features extraction from the skeleton representation and an LSTM 72
5.5 Overall activity confusion matrix in "New Person" setting with the CNN-LSTM model on 140 frames window 83
5.6 Some sampled activities: talking on the phone (up), watching TV (center), ironing and making coffee (bottom). 87

5.7 Normalized confusion matrix of the classification results on our dataset. The label activities are PC, phone, TV, coffee, iron, couch corresponding respectively to working on PC, talking on the phone, watching TV, making a coffee, ironing, talking on couch with another person. 90

5.8 The frame video shows a child with his skeleton joints recognized by OpenPose [8]. 92

6.1 Average execution time for the *find user* service 106

6.2 Average execution time for the *play video* and *play music* service . . 106

6.3 Average execution time for the *check medicine* and *remind medicine* task 107

6.4 Number of personalized and random workflows for each patient . . 108

Acronyms

1NN One-Nearest-Neighbor.

ADL Activity of Daily Living.

AI Artificial Intelligence.

AJAX Asynchronous JavaScript and XML.

ANN Artificial Neural Network.

ANX Anxiety.

API Application Program Interface.

AR Activity recognition.

ATT Attitude.

BLE Bluetooth Low Energy.

BPTT Back-Propagation Through Time.

CA Cronbach Alpha.

CAD-60 Cornell Activity Dataset 60.

CDR Clinical Dementia Rating.

CNN Convolutional Neural Network.

DAA Daily Assistive Action.

DAG Direct Acyclic Graph.

DAW Daily Assistive Workflow.

DAWG Daily Assistive Workflow Generator.

DBMM Dynamic Bayesian Mixture Model.

DMMs Depth Motion Maps.

DNN Deep Neural Network.

DTM Dynamic Time Warping.

EU European Union.

FC Facilitating Conditions.

FP7 7th Framework Programme.

GWR Growing When Required.

HAR Human Activity Recognition.

HCI Human-Computer Interaction.

HMM Hidden Markov Model.

HR Heart Rate.

HRI Human-Robot Interaction.

HTN Hierarchical Task Network.

IAD Alzheimer's disease.

iADLs instrumental Activities of Daily Living.

IoT Internet of Things.

ITEE Information Technologies and ElectricalEngineering.

ITU Intention to Use.

JAX-RS Java API for RESTfull.

JAX-WS Java API for XML Web Services.

JSON JavaScript Object Notation.

kNN k-Nearest Neighbor.

LOO Leave One Out.

LSTM Long short-term memory.

MCI Mild Cognitive Impairment.

MHI Motion History Image.

MIUR Ministry of Education, University and Research.

MMSE Mini Mental State Examination.

Neo-Pi-3 Neo Personality Inventory - 3.

OCCI Open Cloud Computing interface.

OMCRI Open Mobile Cloud Robotics Interface.

PAD Perceived Adaptability.

PENJ Perceived Enjoyment.

PEOU Perceived Ease of Use.

PS Perceived Sociability.

PU Perceived Usefulness.

RaaS Robot as a Service.

RELU Rectified Linear Unit.

REST Representational State Transfer.

RNN Recurrent Neural Network.

ROS Robot Operating System.

RSSI Receive Signal Strength Indicator.

SAR Socially Assistive Robotics.

SDK Software Development Kit.

SI Social Influence.

SOA Service-Oriented Architecture.

SOAP Simple Object Access Protocol.

SP Social Presence.

SSC Systemic Sclerosis.

ST-GCN Spatial Temporal Graph Convolutional Networks.

STIP Spatial-Temporal Interest Point.

SVM Support Vector Machine.

TR Trust.

UPA4SAR User-centered Profiling and Adaptation for Socially Assistive Robotics.

UTAUT Unified Theory of Acceptance and Use of Technology.

VPL Microsoft Visual Programming Language.

XML eXtensible Markup Language.

Chapter 1

Introduction

As our life expectancies steadily increase thanks to the giant strides made in medicine, the elderly population makes up an increasingly large percentage of the population worldwide. Advances in medicine are joined by advances in technology, spreading more and more smart devices and sensors that are now part of our daily lives. Technology can be an enabling tool for elderly people by improving their life. In recent years, Socially Assistive Robotics (SAR) is playing an important role in research for the development of technologies that can improve the quality of life of older people. In particular, I am talking about people with dementia or even lonely people who do not take good care of themselves during daily life, needing support from family or caregivers. Enhance their autonomy would require fewer interventions by family members or caregivers. Particularly in the case of elderly people suffering from dementia, such as Alzheimer's disease, which causes a progressive loss of memory and a worsening of cognitive functions. This disease slowly leads to not recognizing loved ones or feeling disoriented and confused not knowing where the elderly person is or losing short-term memory, thus putting the elderly person at a serious risk of safety. These are just some aspects of this disease that affects not only the elderly but also the family or caregivers who take care of them. SAR applications allow us to autonomously monitor elderly people through a robot that provides information about the patient during daily life checking possible anomalies that are then reported to caregivers. The robot

can be a monitoring tool but also a cognitive and physical stimulus for the patient allowing to make up for memory problems with reminders.

This work has been supported by MIUR within the research project PRIN2015 "User-centered Profiling and Adaptation for Socially Assistive Robotics - UPA4SAR". The main topic elaborated during the master thesis period was the recognition of daily activities performed at home by elderly people with dementia for home monitoring [28]. This first work was the basis for the continuation of studies and research pursued during the PhD ITEE (Information Technologies and Electrical Engineering) carried out at the University of Naples Federico II. The project that funded this work, UPA4SAR, aims to provide a low cost solution, usable by most people, but that is also modular, robust and scalable, to allow a greater dissemination and easier maintenance and expansion of functionality, adapting it to the possible needs of patients. The robotic system to be implemented for elderly care must be easily applicable to any hardware, environment, and functional needs to best accommodate customizations. A pivotal node of such a system for elderly care is acceptance. To improve the acceptability [73] [62] of the system, it must be as unobtrusive as possible, be able to adapt to the user's preferences and allow for pleasant social interaction.

1.1 Problem Definition

The objective of the UPA4SAR project is the creation of a robotic system for monitoring and entertaining elderly people with dementia based on the personalization of the services offered by the robot for greater user acceptance of the system. The project involves profiling the user's individual ADL (Activity of Daily Living) related abilities, cognitive status, and personality. Based on the information collected, it will be possible to plan a set of personalized monitoring activities that, adapted to the current situation, will make it possible to instruct the robot to perform the most appropriate behaviours. The robot behaviours will also take into account the user's preferences derived from his/her profile. There are really many aspects of such a large project that required many heterogeneous

figures to collaborate and cooperate with the intent to improve the life expectancy of the elderly, especially those subject with senile dementia such as Alzheimer's.

For increased and rapid deployment of assistive robots, the hardware of the robotic system should be at a low cost. The robot must perform tasks to monitor a subset of ADLs (Activities of Daily Living) while not distracting the user from their daily activities for greater robot acceptance. Activity recognition is typically based on data from sensors deployed in the surrounding environment. In this project, the robot is the core of the system and is used as an active sensor to monitor the activities performed in the home by the user to avoid adding invasive and often expensive sensors in the environment thus increasing the uptake of this application. The robot must have navigation capabilities within the home environment to be able to monitor activities using sensors for user and context detection. The robot must give the ability to provide cognitive support with reminders and notifications but also provide stimulation through music, videos and games.

The robot must be able to communicate with other devices such as smartband and smartphone. The smartband is a low cost wearable sensor useful to detect the vital functions of the user and his pose (i.e. standing, sitting, walking, running, working on computer, climbing stairs) but also to identify an approximate distance between the robot and the user through Bluetooth communication and the RSSI (Receive Signal Strength Indicator) value. The smartphone instead is a useful device to be able to communicate with a family member or a caregiver, both for social interactions and for possible alarms detected and launched by the robotic system. Other useful devices for the navigation of the robot and the identification of rooms within the home environment are the iBeacons. In order to recognize and identify subjects within the home, the robot must be able to run Artificial Intelligence (AI) algorithms.

For a better acceptance, the behavior of the robot must be able to adapt to the user's needs and cognitive state, a crucial aspect for a real integration in daily life. All the services offered by the robot must be managed automatically

by a scheduler that minimizes the intervention of technicians. In particular, the robot will have to assist the elderly in their home environment in total autonomy, minimizing possible interventions by technicians. The planned experimentation is within the homes of elderly people over 50 years old with different cognitive states, i.e. affected by subjective memory disorder, affected by mild cognitive impairment, affected by Alzheimer's dementia in mild to moderate stage (Mini Mental State Examination, MMSE, greater than 16, Clinical Dementia Rating Score (CDR): 1-2).

1.2 Our Solution

Our solution includes the analysis and implementation of a service-based robotic architecture consisting of several modules and the implementation of an algorithm for monitoring ADLs (Activities of Daily Living). The project UPA4SAR was developed in several phases, from the analysis of the requirements to the experimentation in the homes of the elderly, requiring the approval of an ethics committee. Through the collaboration of various professional figures, it was possible to implement and test the robotic system into the homes of the elderly who offered themselves for experimentation. Doctors and psychologists of the project team were dedicated to finding elderly people with the right requirements for field testing, collecting psychological questionnaires and user preferences in order to profile the subjects and adapt the robotic system to the needs and personality of the patients. During the selection of patients, I worked in parallel on the functional analysis of the system and then moved on to implementation and testing in the laboratory. Several experiments were performed with elderly people who volunteered for interaction with a robot that included topics such as proxemics and adaptation to personality traits [72], distraction created by the robot while the patient performs activities of daily living [68] [71], robot acceptance [16], the robot's approach and interaction with people [70]. I must then consider the elderly monitoring functionality that is based on the recognition of daily activities performed within their home, the so-called ADLs (Activity of Daily Living). The

field of activity recognition, or AR, requires the development of a model trained on registered instances to be able to recognize possible new instances performed by the user [28]. In addition, the recognition of possible deviations from the activity being performed is also fundamental, in order to be able to verify the correct execution in the event of a possible danger that could harm the elderly person [67]. In order to lower the cost of the robotic application, with the intent of spreading this solution to all seniors' homes, we did not use environmental sensors such as external cameras. In most cases, volunteers' homes are not smart homes and do not have a centralized system of sensors for the home. Therefore, the onboard sensors of the robot, which is a proactive device due to its navigation and sensor capabilities, are exploited. For experimentation, we opted to use an assistive robot marketed at a low cost. The robot in question is Sanbot Elf, a cloud-enabled intelligent service robot developed by Qihan Technology Co. Ltd. To easily expand the functionality of the system, for easy code maintenance, and to make it scalable, I opted for a microservice-based architecture that communicates through an event-driven communication framework.

In summary, the main features of our approach are the following:

- the proposed robotic application is low-cost to encourage its deployment
- no external environmental sensors are used such as environmental cameras inside a smart home
- the robot becomes a proactive device due to its sensoristics and navigation capabilities
- a deep neural network model classifies the activities of daily living (ADLs) to monitor the person
- the architecture of the robotic system is modular and scalable because the functionalities have been implemented as microservices
- the communication between microservices is based on an event-based architecture

- the robotic application is performed completely autonomously, without the intervention of technicians, inside the homes of the elderly during the experimentation.
- the robot provides several assistance services such as notifications for taking medicines, entertainment suggestions through multimedia contents that allow to stimulate physically and digitally the elderly people
- the behavior of the robot adapts to the person's preferences to improve acceptance of the robotic system.

1.3 Innovative Aspects

Over the past decade, many projects for robotic assistance of elderly people have been completed or are currently under development. One of the innovative goals achieved that differentiate us from other approaches is the development of a software architecture of the robotic application based on microservices. While projects such as "Accompany" [74] or "SocialRobot" [64] are based on ROS (Robot Operating System), I used the latest technologies in web development on the cloud to speed up the implementation of the robotic application. This allowed us to make the robotic application modular, scalable and easily integrated into all hardware devices. I used "Nodejs" for orchestration and execution of various modules written in various programming languages. Moreover, these modules can be run in containers like Docker [5] that use virtualization to avoid conflicts between dependency libraries. The isolation of the components allows us to minimize possible system interruptions. Communication between system services is done through the "SocketI.IO" library, making the architecture event-based. The architecture initially developed in [28] [67] has been thoroughly tested and found to be a system capable of working in real time without the need for ad-hoc feature extraction. Moreover, this architecture has been trained on the dataset sampled in our lab with the social robot Pepper but also in other different contexts such as gesture recognition. We recorded some activities of interest from elderly volunteers using

the Pepper social robot made by SoftBank Robotics. During these experiments on the robot's approach towards people [27] and distraction during their non-interactive tasks [68] we sampled videos of the activities performed by the elderly using Pepper's front-facing camera. Using this data, I trained a deep neural network model and obtained a good accuracy result. Finally, another goal achieved is the use of a low-cost, fully autonomous robotics system that adapts to the user's needs. One of the reasons why we still don't have a global deployment of assistive robots is precisely the excessive cost. Not everyone can afford a social robot at home. Mostly, they are very common on cruise ships or in shopping malls. Lowering the overall cost of the entire robotic system, making it fully autonomous and adaptable to the user's profiles would allow a rapid diffusion of this application within the homes of the elderly, encouraging improvement and research in this field.

In summary, there are three innovative aspects of the proposed robotic application:

- the implementation of a modular robotic architecture based on micro services.
- the training of a deep neural network model with a dataset sampled during experiments with elderly volunteers in our laboratory
- the low-cost experimentation in the homes of the elderly with a fully autonomous robotic system that adapts to the needs and cognitive profile of the person

1.4 Thesis' Contribution

I would like to emphasize that this thesis work is about my contribution offered to the UPA4SAR project. In particular, I was involved in

- analysis and implementation of the robotic system with all the services offered

- recognition of Activities of Daily Living
- experimentation in the homes of the elderly
- final analysis of the experimentation.

1.5 Structure of the Thesis

In chapter 2, I introduce the state of the art of the three main topics pertaining to this thesis. In section 2.1, I present some assistive robot projects for elderly people tested in controlled environments and smart homes. In Section 2.2, I introduce the Service-Oriented Architecture (SOA) approaches that are at the base of our project. Finally, in section 2.3, I discuss some Activity of Daily Living (ADL) classification approaches. In chapter 3, I discuss the architecture of the proposed robotic system, starting from the functional requirements described in section 3.1 to the discussion in section 3.2 of the architecture developed and tested during field experimentation in elderly homes. In chapter 4, I first consider an overview of the software analysis in section 4.1 and provide a more detailed description of some of the implemented system components in the section 4.2 and 4.3. Specifically, these components are divided into three main groups, which perform different types of functionality, respectively: A) Monitoring Services in section 4.4, which include activity recognition via a wearable device or via camera using pose/skeletal recognition, emotion recognition, and disengagement; B) Navigation Services in section 4.5, for user search and approach; and C) Interaction Services in section 4.6, for speech recognition and synthesis using multimodal user interaction. In section 4.7 I discuss about the communication between the services and in section 4.8 I show the tests conducted on the response times of person finder services within the simulated home environment. In chapter 5 I introduce the approaches used during experimentation for activity recognition based on a deep neural network described in section 5.1. The proposed models were first trained on the public CAD-60 dataset in section 5.2. I then described a deep model, in section 5.3, trained on a dataset recorded in our lab during experiments with elderly volunteers. An

approach used as part of a collaboration with the Sheffield Hallam University to recognize the gestures of autistic children mimicking the gestures of a robot is described in section 5.4. In chapter 6, I show and discuss the final results of the experimentation conducted within the homes of elderly volunteers. Finally, in chapter 7, I summarize and show some conclusions of the research experiments described in this thesis.

1.6 Publications

- Rossi, S., Bove, L., Di Martino, S., & Ercolano, G. (2018, November). A Two-Step Framework for Novelty Detection in Activities of Daily Living. In International Conference on Social Robotics (pp. 329-339). Springer, Cham.

- Rossi, S., Ercolano, G., Raggioli, L., Savino, E., & Ruocco, M. (2018, August). The Disappearing Robot: An Analysis of Disengagement and Distraction During Non-Interactive Tasks. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) (pp. 522-527). IEEE.

- Ercolano, G., Raggioli, L., Leone, E., Ruocco, M., Savino, E., & Rossi, S. (2018, August). Seeking and Approaching Users in Domestic Environments: Testing a Reactive Approach on Two Commercial Robots. In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) (pp. 808-813). IEEE.

- Rossi, S., Ercolano, G., Raggioli, L., Valentino, M., & Di Napoli, C. (2018). A Framework for Personalized and Adaptive Socially Assistive Robotics. In WOA (pp. 90-95).

- Rossi, S., Santangelo, G., Ruocco, M., Ercolano, G., Raggioli, L., & Savino, E. (2018, March). Evaluating distraction and disengagement for non-interactive robot tasks: A pilot study. In Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (pp. 223-224).

- Di Napoli, C., Del Grosso, E., Ercolano, G., Garramone, F., Salvatore, E., Santangelo, G., & Rossi, S. (2019). Assessing Usability of a Robotic-Based AAL System: A Pilot Study with Dementia Patients. In WOA (pp. 59-64).

- Ercolano, G., Lambiase, P. D., Leone, E., Raggioli, L., Trepiccione, D., & Rossil, S. (2019, October). Socially Assistive Robot's Behaviors using Microservices. In 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (pp. 1-6). IEEE.

- (accepted, in print) Ercolano, G., Rossi, S. Combining CNN and LSTM for Activity of Daily Living Recognition with a 3D Matrix Skeleton Representation. Intelligent Service Robotics – Springer

- (submitted) Di Napoli, C., Ercolano, G., Rossi, S., Personalized Home-care Support for the Elderly Learn by Experience with a Social Robot at Home. User Modeling and User-Adapted Interaction – Springer

- (in submission) Ercolano, G., Rossi, S., Conti, D., Di Nuovo, A., Gross Motor Imitation skills in Autism Spectrum Disorder with Intellectual Disability: gesture recognition using 2D camera. Applied Soft Computing – Springer

Chapter 2

Related Work

In this chapter, I discuss the related work of the three main topics addressed during my PhD. As my thesis research is related to the MIUR UPA4SAR project, the main topic is the daily care of the elderly through the use of social robots, especially elderly people with dementia such as Alzheimer's disease. In the future, I expect social care robots to be deployed in every home, not just for seniors, and based on a cloud-based service architecture: Robot as a Service (RaaS). Among the various monitoring services for the care of the elderly, the Activity of Daily Living (ADL) classification has been at the heart of my study path.

In section 2.1, I present some assistive robot projects for elderly people tested in controlled environments and smart homes. In section 2.2, I introduce the Service-Oriented Architecture (SOA) approaches that are at the base of our project. Finally, in section 2.3, I discuss some Activity of Daily Living (ADL) classification approaches, considering the different models used to deal with the Activity Recognition problems.

2.1 Assistive Robot for Elderly Home Care

Socially Assistive robotics is a branch of robotics that aims to assist the user by following their needs, through observation of the user's behavior and health conditions. To ensure the user's health status and safety, the robot monitors the

person on a daily basis, reminding them to eat, take medications, or entertains them through activities to keep the assisted person both physically and intellectually active. Some of the main topics addressed in this field include assessing and increasing the acceptance of these systems for a daily support to elderly people, allowing them to be active and independent at home.

For the development of our robotic platform, we have been inspired by projects previously proposed for the care of the elderly. There are several projects funded by European Union (EU) that have seen the cooperation of universities, companies and professional figures of various fields for the realization of the same. Our project is based on the cooperation of various figures including doctors, psychologists, professors and engineers and various research institutions: University of Naples Federico II - Department of Electrical Engineering and Information Technology, National Research Council - Institute of Parallel Computation and High Performance Networks (ICAR - CNR), University of Naples Federico II - Department of Neuroscience, Reproductive Sciences and Odontostomatology, University of Campania L. Vanvitelli - Department of Psychology.

The aim of this project is the development of a mobile robotic platform able to interact with the elderly for physical and cognitive stimulation, monitoring and entertainment. It focuses primarily on profiling and adapting to the personal needs and preferences of the elderly. I present hereafter other projects following the same lines of research which I have compared with our own.

In the European Union (EU), the 7th Framework Programme (FP7) for research funded several projects in social robotics. In these projects, a robotic companion was developed to interact with humans in a smart home environment for cognitive stimulation and therapy management of patients with cognitive disabilities and for monitoring the activities of the users.

In the "GiraffPlus" [13] project, the user leaned on a telepresence robot to communicate with relatives or caregivers. Projects based primarily on an autonomous social robot that delivers services to the user, are "Accompany [74]", "Robot-Era" [6], "Mobiserv" [54], Hobbit [30] [65].

In the "Accompany" [74] project, the Care-O-Bot 3 was developed for more complex interactions with the elderly people, providing services to facilitate independent living at home, assisting the user in the daily tasks. The authors created an ontology of the robotic home, consisting of sensors, locations, objects, people, the robot, and the robot's behaviors. This ontology has been instantiated in a MySQL database and is used to model and test the robotic system, representing the various artifacts in the ontology not as separate parts but as units belonging to the same environment. The robot's navigation is entrusted to ROS using the robot's range-finders lasers, allowing a real-time map to be updated and obstacles to be avoided. The house contains about 50 sensors, for example sensors that detect when doors are opened or appliances are turned on, and which communicate with the system via ROS messages. In addition, there are cameras mounted on the ceiling and external sensors which detect the weather. Robot behaviors are automatically generated through a template consisting of pre-conditions followed by the execution of robot actions and post-conditions that update the system state. The scheduling system used is the open source scheduler HTN (Hierarchical Task Network) (SHOP 2 [55]) that encodes each scheduling domain via a lisp-like syntax, using a scheduling priority assigned to each behavior. If each precondition of a behavior is met then the behavior is executed, following the assigned priority order. This project allows for customization of behaviors by both experts and non-experts, via simple interfaces. Even the elderly person can issue commands via the tablet. However, there appears to be no mention of clinical planning of elderly care activities by caregivers. In addition, the system requires an expensive home automation environment with a large robot that cannot be adapted to any home environment.

The "Robot-Era" [6] project integrated advanced robotic services in a smart home environment, in real world conditions, cooperating with elderly people to improve their life and facilitate independent living. In this project, they highlight the use of cloud systems for service delivery and the use of sensors to locate the person within the apartments, via invasive devices such as mobile radios worn by

the users.

In the "Mobiserv" [54] project, the Authors created an intelligent system with a robot and smart sensors for assisting older people. They focused on health monitoring (using wearable sensors), safety and nutrition support. For example, the implemented robot can remind the user to drink or eat something, to do physical exercises or a puzzle game, and - through sensors - it controls gas, water, windows and doors, checking if they are open. All these possible behaviors and scenarios of the robot have been assigned to each user to make the adaptation and customization of the system flexible, according to personal needs.

The robot developed in the "Hobbit" [30][65] project can learn, bring and pick up objects. It provides entertainment functionalities, such as games, exercise, films, music and books to keep the user cognitively and physically active. Importantly, it can detect warning situations. The robot is able to recognize, carry, and pick up objects as in the "Accompany" [74] project. This type of design allows for basic robot customization through user choice of sound volume, robot speed, and robot voice gender, but does not allow for more advanced customization on the schedule and services offered by the robot. All the projects described above, proposed a robotic platform relying on a modular architecture.

This concept is also taken up in the European collaborative project, called "SocialRobot" [64], where the Authors developed a mobile robotic platform by integrating it with virtual social assistance technology to meet the personal needs of the elderly by adapting during the aging process. Their ultimate goal is to maintain users' self-esteem while managing their daily routines. The proposed platform consists of a modular system to support caregivers, family members, friends, who follow the elderly during their daily tasks. The innovation in this project is adaptation in human-robot interactions, for example through emotion recognition and empathic interaction, and adaptation to daily events. The proposed architecture [64] is based on a Workflow Engine that interprets service requests described by XML messages. The services are then called through ROS (Robot Operating System) services. This kind of approach has been the inspiration for our project,

which differs in technologies and architecture used. In [63], related to the "Social-Robot" project, the Authors tested their mobile robot platform in a care center during a week-long pilot. It targets elderly with mild physical or psychological issues, offering a set of smart services providing advanced user-robot interaction: for instance, face recognition, emotion recognition, word spotting, navigation to a specific place/room, approach to the person, monitoring, docking/undocking, speech synthesis. They focused on security, autonomy, privacy and safety, managing the daily routine with a modular designed platform that mainly supports caregivers and family.

The "Robot-Era" [6] and "SocialRobot" [63] projects followed the principles of a Service-Oriented Architecture (SOA) with a hierarchical approach where each complex service is the composition of simpler operations. Their modular service robot architecture promotes scalability and layer abstraction exploiting high-level service personalization according to the user preferences and habits. In [6], the Authors proposed the improvement of the performance of the robot behaviors by using a cloud robotics paradigm, with the Robot as a Service (RaaS) architecture that provides scalability, elasticity, computational capabilities and much more, enabling to offload complex processing, user and environment information sharing. However, the tests were conducted in a home automation environment and a care facility, both controlled environments, focusing on the latency of the services offered by the cloud and the localization of the user using invasive sensors which aims at improving the acceptance of the robotic system without the use of sensors inside the house or invasive wearable sensors. The most interesting works in literature that considered the testing of the robot platforms in real environments, like home care or controlled apartments, with a considerable number of patients and with medium to long term periods, are proposed below.

In another early project related to the European FP7 project "Companion-Able" (2008-2012) [76], the Authors developed a companion robot for assistance by combining a mobile robot with a smart home used for testing. Volunteers tested the system by freely using the robot for two days. Services provided by the robot

include reminders of appointments, predefined or added by the user or caregiver, frequent recommendations for specific activities provided by the caregiver, video calls with family and friends, and cognitive stimulation games to monitor for cognitive impairment as well. In addition, the robot they developed can be controlled via displays located within the smart environment or via voice command to call the robot for instance. In total, they conducted 12 days of testing, providing an enjoyable experience for the volunteers, elderly people with cognitive impairments, and their partners. Projects such as "CompanionAble" present experimentation in appropriately prepared apartments or in care centers used for the integration of IoT devices in order to help the robot to provide more reliable services. Using IoT devices to create a smart home would increase the cost of the project by requiring structural interventions within the seniors' homes.

The same robot was proposed again for the German research project "SERROGA" (2012-2015) still in the field of home care for the elderly, with the task of keeping the elderly physically and mentally active. In [33] this project is presented with a new approach to allow a quantitative description and evaluate the complexity of navigation in the apartments, so that they can be compared for functional testing under real-world conditions. 12 project staff and elderly apartments were tested, with 9 elderly people, aged from 68 to 92 years. In total, testing in private apartments of the project staff and the elderly lasted 7 days, and in the elderly apartments lasted 16 days, where the robot interacted with the elderly in complete autonomy, without supervision. The testing of the experiment with the robot alone, without any person supervising, lasted a maximum of three days. The elderly ultimately enjoyed the functionalities of the robot and also made emotional connections with it. The hypothesis formulated by the Authors is confirmed by the results: robots provide psycho-social and instrumental advantages over computers, tablets, or televisions due to their physical presence, mobility, and social interaction capabilities. These capabilities help the elderly to overcome loneliness and improve people's psycho-physical well-being.

The German research project "SYMPARTNER" (2015-2018) presented in [34]

aimed to develop a mobile domestic robot that can stimulate the elderly. In [34], they did not consider user customization and profiling a fundamental requirement. The authors consider that it would be necessary to allow the system to adapt to the user's needs and preferences as possible future developments of the project. Therefore, The architecture of the robotic system, its components, and essential behaviors are presented. A long-term study carried out from January to June 2018, with two autonomous robots used in 20 elderly households, in complete autonomy, without supervision, highlighted technical aspects on the suitability and robustness of the robot, usability, familiarization and acceptance of the robot with its users. The tests lasted to 20 weeks, with elderly people aged 62 to 94 years. The services offered by the robot during the trial worked autonomously for the majority of cases, with local navigation problems occurring in only a few cases. This project highlights the importance in the implementation of navigation methods, interaction and other services offered. The level of satisfaction and the expectations of users on the individual support services offered by the robot should be further investigated. Even if the long-term experimentation has given reliable results, the robot platform proposed in [34] presents some possible improvements: the touch screen, for instance, should allow an easier access when people are standing; the proposed robot is not able to overcome thresholds and higher carpets, along with other irregularities.

In [35], an adaptive approach is proposed that allows scheduling of robot activities that can be modified even by lay users such as caregivers of elderly people prone to Alzheimer's disease (IAD). A caregiver can customize the default care protocol through a survey-style questionnaire that contains binary or multiple-choice questions and a small number of open-ended questions to provide detailed instructions. The responses are then processed to automatically generate the new schedule. Open-ended responses must be written by the caregiver in a predefined language structure for the generator to extrapolate predicates or operators. This type of approach allows for continued adaptation of the protocol by the caregiver. Through simple questions posed to caregivers, it is possible to customize a care

protocol on an ongoing basis by modifying the robot's high-level daily planning. Two protocols for the care of older adults with dementia have been implemented to evaluate this approach. The robot's planning also relies on sensors set up in the home to detect activities regarding the care protocol. The proposed framework was evaluated with the help of caregivers who redesigned the care protocols based on the preferences of the testers, 8 IAD patients.

Regarding the functional requirements, the following study is very interesting for the development of an affordable service-oriented robotics platform. In [40], a study of the requirements for developing an affordable mobile robot is presented. This study is about researching the needs and requirements of the elderly in a nursing home. Three groups of subjects were composed for this study: elderly, clinicians, and caregivers. Their goal is to define the system requirements needed by a robotic system for daily care, using a low-cost robot with affordable prices. The three groups identified an order of priority to various services. Thirty-six unique functions with the highest priority were then sketched from their evaluations. The most important services that seniors desired in a low-cost robot related to instrumental activities (iADLs), leisure activities, making their preferences known, and the ability to socialize. In contrast, physicians and caregivers prioritized reminders and adherence to care plans, monitoring the health and safety of the elderly. Of course, a low-priced robot requires trade-offs in design and service robots provide some but not all desired services. This study suggests that the best elderly care robot promotes connectivity between the elderly and the community, entertainment, safety, health care, and aids in simple recovery activities.

Finally, in a recent project [14] funded by European Union (EU) under the Horizon 2020 program, the Authors highlight the innovations of perception and interaction between the mobile robot developed by the Authors and the users, for in-home assistance to elderly people with mild cognitive impairment. In particular, they developed a robotic system capable of mapping everyday objects in the home through RFIDs and able to monitor users with non-invasive tools. The phys-

iological monitoring module is based on the Fourier analysis of the temperature of the forehead and nose, thus being able to detect the temperature, respiration, heart rate while the elderly entertains himself with cognitive games or other interaction tools of the robot. For the detection of physiological functions, it is essential that the face is visible to the robot's camera. This platform allows for more dynamic and flexible user monitoring and object mapping, adapting even in cluttered environments, thus eliminating the need for fixed sensor installations and configurations within the home. Our project is based on the adaptation and profiling of the person based on several psychological tests performed by the team of psychologists. Our experimentation aim to verify the degree of comfort and acceptance of the users with respect to the robotic system based on two protocols used for planning: a personalized day planning and a random one. One of the key goals of our project was to experiment in real-world settings, specifically in the homes of the elderly and not in assisted living facilities. It was carried out on 7 volunteers for an average duration of two weeks for each. The robot was left in complete autonomy without the intervention of operators, without connection to the Internet to preserve the privacy of the elderly and their family. We avoided the use of environmental sensor like cameras to limit the cost of the platform and to decrease invasiveness.

2.2 Service-Oriented Architecture (SOA)

The introduction of Industry 4.0, 5G, IoT devices, smart devices (smartphones, tablets, smartwatches, smartbands), home automation, voice assistants, and robotics have made service-oriented architecture (SOA), edge computing, and cloud computing increasingly dominant. Hence, many researchers started to use such approach also for designing robotics applications.

For example, in [9], the design and evaluation of a Robot as a Service (RaaS) prototype is presented. The proposed platform follows the participatory principles of Web 2.0, sharing the source code to allow developers modifications and additions to the RaaS on Windows and Linux operating systems and on Atom and Core 2

Duo architectures. To make the customization of the services accessible also to high schools, in addition to the possibility to develop services with languages such as Java and C#, it supports the Microsoft Visual Programming Language (VPL) to be able to graphically compose the services, allowing the graphical composition based on Robotics Developer Studio. This design is very flexible, easily portable to different systems, and has been tested in various experiments. The Authors also proposed a robotics starter kit for dissemination in high schools and leveraged this design to create training courses in robotic programming [10] to attract student interest and teach robotic service oriented programming (RaaS).

In [7] the Authors propose an early stage approach based on ROS (Robot Operating System) which is an operating system for creating robotic platforms widely used in both research and industrial sectors, providing libraries and tools to help developers create robotic applications. Their approach is based on the use of SOA technologies, i.e. service-oriented architectures, and web services. The basis of the platform is the communication between the client, i.e. the robot, and the provider that provides various services through the web with the help of Cloud architecture. It differs from our project for the use of ROS, voluntarily not used because I considered it as an additional module and level of communication with other robotic platforms, keeping our robotic platform at a lower level of communication, to reduce the latency and not add more complexity. ROS remains one of the most widely used research tools in our lab as well, making module (node) implementations very easy and fast. Instead, I looked for a viable alternative for a fast and robust implementation that would safely allow communication and integration with ROS.

Another similar ROS-based approach is proposed in [1]. The Authors considered the ROSJAVA client Application Program Interface (API) for representing meta-models using web services implementing two communication protocols: JAX-WS (Java API for XML Web Services) and JAX-RS (Java API for RESTfull). This allowed the development of REST (Representational State Transfer) interfaces based on XML (eXtensible Markup Language) documents using SOAP

(Simple Object Access Protocol) messages. The Authors' goal is to control service robots through these Web interfaces, leveraging the computational power of Cloud computing. The robot then acts as a client requesting computational resources from the cloud for various artificial intelligence services that are often difficult to perform on machines with low computational capacity such as social robots or drones. The system proposed in [1] was then tested for drone surveillance.

Through a unified interface to easily manage heterogeneous mobile robots remotely, the Authors of [53] propose their own platform called Open Mobile Cloud Robotics Interface (OMCRI) that leverages the RaaS architecture. This platform is an extension of the Open Cloud Computing interface (OCCI) standard and leverages the OCCIware tools, testing them on three mobile robots: Lego Mindstorm NXT, Turtlebot and Parrot AR.Drone. This platform is also modular and extensible, but is dependent on the Open Cloud Computing Interface (OCCI), an open source project that requires commitment from developers to maintain.

Other projects I have already considered in section 2.1 that used service-oriented architecture (SOA) are the projects "Robot-Era" [6] and "SocialRobot" [63]. Each complex service is composed of elementary services or operations to form a hierarchical service structure. Such a modular architecture allows the robotic system to scale by abstracting services at various levels for easy parameterization and customization of services. Indeed, the services must adapt to the preferences and habits of the elderly person. A further improvement is the use of decentralized software architectures such as edge computing or cloud computing: an example is the work [6]. In [6] the authors want to improve the performance of the offered services by exploiting the Robot as a Service (RaaS) architecture that offers scalability, elasticity and computational power, being able to process all the information of the user and of the surrounding environment extrapolated from IoT sensors. A limitation of this type of architecture, however, is the multi-layered security of the architecture over data transmitted over the cloud or privacy that is often violated or, under certain conditions, user information is exploited by companies that take advantage of privacy contracts that are unclear, verbose

or difficult to interpret. In [64] the daily schedule is generated by an engine, called the Workflow Engine. This engine processes requests via XML messages, using ROS (Robot Operating System) services.

Our solution allowed us to easily integrate artificial intelligence modules, such as activity recognition and facial recognition, but most importantly, it facilitated communication with the robot used in the project. It allows the modification and implementation of new services with various programming languages like Python, Javascript, Java. Moreover, the communication of our robotic platform is based on Socket.IO, a library of Node.js. Socket.IO, an event-oriented JavaScript library, enables real-time two-way communication between web clients and servers. Thanks to this library, it is possible to make heterogeneous modules communicate using clients made available in various programming languages. The architecture presented in our work is service-oriented architecture (SOA) with a hierarchical approach. Each complex service is the composition of simpler operations. This type of architecture is also used in projects such as "Robot-Era" [6] and "Social-Robot" [63]. It is a scalable modular architecture that allows the customization of services according to the preferences and habits of the elderly.

2.3 ADL Classification Methods

Human Activity Recognition (HAR) aims to recognize activities or actions and it is a challenging task for human activity classification. In this section, I considered different activity recognition approaches applied on different datasets and in particular on dataset CAD-60. I considered the CAD-60 dataset since it contains people's daily self-care activities (ADLs) and it has been used by us to achieve our goals. The works presented concerns machine learning methods that are currently state of the art. Deep artificial neural network models are used in these works like Long short-term memory (LSTM) and Convolutional Neural Network (CNN), in conjunction with classical machine learning techniques for feature extraction.

In [24], the Authors introduced an approach where the human skeleton data is analyzed by considering five parts (i.e., arms, legs, and torso) and a hierarchical

bidirectional RNN network with a final Long short-term memory layer is deployed to extract features for building a higher-level representation. Subsequently, a fully connected deep LSTM network is proposed in [85] to recognize action with a framework composed of three LSTM and two feed-forward layers, incorporating the co-occurrence regularization into the loss function, so exploring the conjunctions of discriminative joints and different co-occur-rences for several actions. In [47], a deep LSTM framework, based on RNN, is proposed to better localize the start and end of action with a regression module, to automatically extract the features. This joint classification-regression RNN considers the sequence frame by frame and does not require a sliding window approach. A hierarchical approach is also presented in [84], where they propose three exploration fusion methods based on multilayer LSTM. The first LSTM layer takes geometric features computed on the 3D coordinates of the human joints, then the upper LSTM layers investigate into more detail the input features, abstracting into a high level of knowledge. All these approaches are characterized by a deep/hierarchical structure aiming at recognizing high-level features for temporal data. Indeed, in the presented work, a single LSTM layer is used but in combination with CNN, so relying on the possibility to extract spatial dependencies on the skeleton's joints patterns. In [28], we showed that a multi-scaled LSTM approach resulted in slightly lower performances with respect to the proposed one.

Other approaches dealt with the use of CNN for activity recognition. For example, in [23], the skeleton sequence is represented as a matrix concatenating all frames together in chronological order. This allows us to treat the time sequence of joints in a single image that is fed into a CNN model for feature extraction and activity recognition. In [11], the whole images, and not the skeletons, are used to extract joint heatmaps (using CNNs) for each video frame and colorize them using a specific color depending on the relative time. To obtain a fixed-size representation independent from the duration of the video, they aggregate the colorized heatmaps with different methods to obtain the clip-level representation with a fixed dimension. The necessity of compressing temporal date into single images is

overcome by the use of 3D CNN that recognizes spatial-temporal features applying convolutions on a time series of frames [2, 39]. Also in [57], the Authors consider as input for CNN all the skeleton joints of all frames, arranged in a 3D matrix. This 3D matrix has in the first dimension the number of joints, in the second dimension the number of consecutive frames and in the third dimension the 3D coordinates of the joints. Results of the CNN is then combined with an LSTM using a two-stage training strategy that focuses first on CNN training and then on the entire CNN+LSTM method. In our approach, the CNN takes as input each frame of the sequence independently since temporal relationships are deployed at the LSTM layer. In [83], the Authors propose a novel model of dynamics skeletons called Spatial Temporal Graph Convolutional Networks (ST-GCN) tested on Kinetics and NTU-RGBD datasets. The ST-GCN implementations are different from 2D or 3D CNN since the temporal properties of the skeleton are kept together as in a graph. It follows the similar implementation of graph convolution [43]. In [49], the Authors consider the action recognition and the human pose estimation as one problem that they solve with a multi-task CNN. The human pose estimation is composed by a CNN with one entry flow and K prediction blocks to estimate both the 2D and the 3D pose by volumetric heat maps. Appearance-based recognition relies on local visual features considering also the objects used during the performed action. The results are combined to estimate the action. Finally, in [46], the Authors use a combination of CNN and LSTM to extract spatio-temporal information, but, differently from our approach by merging the individual scores obtained from the CNN and the LSTM. Also in this case, contrary to the method proposed by us, they consider all the joints of the skeleton, extrapolating also other information of distance and trajectory between the joints and the poses. The 3 LSTM models take in input the real positions, distances between joints, distances between joints and lines, while the 7 CNN models take in input the joint distances maps and the joint trajectories maps in time to generate color image to be fed into a CNN mode. The innovation in our proposed approach compared to similar works presented so far is in proposing a new spatial representation of the

features of human pose.

Different approaches are presented in the literature that are evaluated by the use of the CAD-60 dataset. These approaches are mainly characterized by different features extraction initial processes. In [12], for example, a k-means clustering algorithm computes the "key poses" to describe the activity for each sequence with K centroids that composed the features vector. In [78], the key poses are identified by recognizing poses with the kinetic energy close to zero to perform a sequence segmentation. This approach is shown to be robust respect to the temporal stretching of an action. In [41], the fusion of 5-CNN is proposed for activity recognition, using Motion History Image (MHI), Depth Motion Maps (DMMs) (Front, Side, Top), and skeleton images (an image representation of the skeleton joints) as input. Each different type of data is trained on a different CNN and the softmax scores are fused to classify the activity. In [29], the distances and motion features (evaluated as the distances between the initial position of a joint and the position in the following frames) form a total of 14 features that characterize the 12 activities of the dataset. A Dynamic Bayesian Mixture Model (DBMM) is proposed to classify the activity considering the temporal information. Depth-based action recognition is evaluated in [86] using the spatial-temporal interest point (STIP) with the combination of different interest point detectors and descriptors. The SVM classifiers are used to detect the activity. A neurobiologically-motivated approach is presented in [61] to recognize action in real-time with the Growing When Required (GWR) networks. The GWR network is a set of neurons that dynamically change their topological structure according to the input creating new neurons with different weights. The architecture proposed is a two-stream hierarchy of GWR networks that can learn spatio-temporal dependencies processing in parallel the pose and motion features extracted from video sequences.

In a recent work [48], Liu et al. proposed a classical machine learning technique, selecting the features from the skeleton data. First, they pre-processed the skeleton data denoising, transforming, and normalizing the pose. Then, they considered the position, the velocity, and the acceleration of the poses. The recognition method

is a three-step weighted voting process based on k-Nearest Neighbors (kNN). They evaluated their method on MSR-Action3D and CAD-60 datasets, obtaining good results. Currently, this approach is the one obtaining the best performance on precision and recall for CAD-60 considering a whole video sequence. The main difference between our work and [48] is that I use a sliding window solving a totally different problem. The obtained model trained on 140 frames instances can classify activities in real-time on videos of a few seconds. I also tested our model on the whole videos to compare our approach with the others. Unlike the approaches applied on the CAD-60 dataset that select and extract features manually, I propose a deep learning model for automatic feature extraction that uses CNNs to extract spatial dependencies from human poses and LSTMs to extract temporal dependencies between poses.

2.4 Summary

In summary, in this work, I attempt to implement and testing in real world environment a personalized social assistive robot to take care of elderly people. There are a planty of assistive robot projects for elderly people, often tested in controlled environments such as laboratories or smart homes. While many projects have focused on deploying intrusive sensor technologies toward the person's privacy, we seek to bring assistive robots even without in-home sensors or video surveillance cameras as far as lack of Internet access. Being able to develop a robot that works autonomously for several weeks is itself an accomplishment. In the Table 2.1 we show a summary of the main features of interest of the related projects.

The main features considered are the following:

- *Telepresence*: the robot is **only** used in telepresence mode to communicate with relatives and caregivers;
- *Autonomous*: the robot is fully autonomous and offers a range of services for the daily care and entertainment of the person;

Table 2.1: Main features of the assistive robot projects for elderly people.

Project	Telepresence	Autonomous	Environment sensors	Arm
GiraffPlus [13]	X			
Accompany [74]		X	X	X
Robot-Era [6]		X	X	
Mobiserv [54]		X		
Hobbit [30][65]		X		X
SocialRobot [63]		X	X	
CompanionAble [76]		X	X	
Our Project		X		

- *Environment sensors*: the robot has environment sensors in a smart home such as cameras e.g. to locate the person inside the house;
- *Arm*: the robot is able to pick up and carry objects via a robotic arm.

In contrast to other projects, ours aims to achieve rapid deployment in the production phase through affordable cost. For this reason and due to privacy concerns, we did not use intrusive environmental sensors. In addition, a robotic arm to pick up and carry objects would greatly increase the total cost of the system and would require environments suitable for the possible movements of the arm that requires a lot of space.

Regarding the literature regarding Service-Oriented Architecture (SOA), we can summarize the technologies used in the following list:

- RaaS architecture based on principles of Web 2.0 [9]
  - two programming languages: Java and C#
  - it supports the Microsoft Visual Programming Language (VPL) (Robotics Developer Studio)
- early stage approach based on ROS (Robot Operating System) [7]
- ROSJAVA client Application Program Interface (API) [1]
  - meta-models representation using web services

  - two communication protocols: JAX-WS and JAX-RS that use SOAP messages

- modular and extensible platform Open Mobile Cloud Robotics Interface (OMCRI) [53]
  - extension of the Open Cloud Computing interface (OCCI) standard
- the projects "Robot-Era" [6] and "SocialRobot" [63]
  - each complex service is the composition of elementary services
  - modular architecture
  - scale system
  - profile adaptation
- Robot as a Service (RaaS) architecture [6]
  - it offers scalability, elasticity and computational power processing IoT sensors data
  - privacy issues due to exploitment of user information by companies
- daily schedule Workflow Engine [64]
  - it processes XML requests using ROS services
- our project
  - various programming languages for the implementation of services like Python, Javascript, Java
  - Node.js handles the executions of the services
  - Socket.IO, an event-oriented Javascript library, handles the communication between the services
  - easy communication with the robot (using a Socket.io client) that used a Android tablet to manage it

  - the communication system is based on JSON messages that can invoke various services
  - even if the system can run on the cloud as a RaaS architecture, for privacy issues the system runs on a small server, an Intel NUC, in offline mode, thus avoiding security issues due to unauthorized external accesses

In conclusion, regarding the methods of Activity Recognition, we can summarize in the following list the various methodologies I have considered in literature:

- human skeleton data divided in five parts (arms, legs and torso) and hierarchical bidirectional RNN network with final LSTM layer [24]
- fully connected deep LSTM network composed by three LSTM and two feed-forward layers [85]
- a deep LSTM for action segmentation with regression module to automatically extract the features [47]
- a hierarchical approach with three fusion methods based on multilayer LSTM [84]
- multi-scaled LSTM approach [28]
- CNN approach with the skeleton sequence represented as a matrix concatenating all frames [23]
- the whole images are used to extract joint heatmaps using CNNs [11]
- 3D CNN [2, 39]
- skeleton joints arranged in a 3D matrix as input of the CNN [57]
- Spatial Temporal Graph Convolutional Networks (ST-GCN), the temporal properties of the skeleton are kept together as in a graph [83] [43]
- multi-task CNN [49]

- combination of CNN and LSTM to extract spatio-temporal information merging the individual scores from CNN and LSTM [46]

The considered ADLs recognition approaches applied, in particular, to the CAD-60 dataset are the following:

- k-means clustering algorithm computes the "key poses" to describe the activity for each sequence with K centroids that composed the features vector [12]
- the key poses are identified by recognizing poses with the kinetic energy close to zero to perform a sequence segmentation [78]
- fusion of 5-CNN using Motion History Image (MHI), Depth Motion Maps (DMMs) (Front, Side, Top), and skeleton images as input [41]
- the distances and motion features as input and a Dynamic Bayesian Mixture Model (DBMM) as classifier [29]
- spatial-temporal interest point (STIP) with the combination of different interest point detectors and descriptors as input and SVM as classifier [86]
- Growing When Required (GWR) networks learning spatio-temporal dependencies processing in parallel the pose and motion features extracted from video sequences [61]
- position, velocity, and acceleration of the poses as input and three-step weighted voting process based on k-Nearest Neighbors (kNN) as classifier [48]

The approach proposed in this work for ADL recognition will be discussed in detail in the chapter 5.

Chapter 3

Architecture

This chapter discusses the architecture of the proposed robotic system, starting from the functional requirements described in section 3.1 to the discussion in section 3.2 of the architecture developed and tested during field experimentation in elderly homes.

3.1 Functional Requirements

Before starting the experimentation in elderly people's homes with assistive robots, we had to analyze the functional requirements and design a service-based robotic architecture. In this section, I will discuss the functional requirements of the robotic platform for elderly care.

An in-depth study of the functional requirements for the development of an affordable mobile robot is considered in [40]. The authors, in particular, focused on researching the needs and requirements of the elderly, one of the key features of our work, within nursing homes. Three groups of subjects, elders, doctors, and caregivers defined their system requirements by prioritizing various services. It was noted that the elderly prefer a low-cost robot that considers instrumental activities (iADLs), leisure activities, considers their preferences, and is able to socialize with them. Doctors and caregivers, on the other hand, placed a higher priority on reminders and achieving care plans by monitoring the health and maintaining the

safety of the elderly. It must be taken into consideration, however, that a low-cost robot requires tradeoffs in design.

The objectives to be achieved are the implementation of services offered by the robotic system for profiling and adaptivity of the assistance robots. The robot must be able to monitor and assist the user, remembering the various activities to be performed during the day and entertaining him with games, songs and videos customized according to the user's profiling. The experimentation in the users' homes will last two weeks. The robot will not be followed by any operator and will not be in Wizard of Oz mode (technique widely used in human-robot interaction (HRI)), but will operate in complete autonomy. The planning of the robot behaviors will have to follow the user's habits. Fundamental importance will be an initial interview with the user to get all the necessary information for the adaptation and planning of the behavior of the robot with respect to the preferences and schedules of daily routine activities of the user.

Four types of IoT devices will be distinguished:

- the robot that will have to monitor through its own sensors (or through a sensor wearable by the user) the activities, emotions and vitals of the user, reminding the user to take his medication, to eat, to distract himself through a game, music or an entertainment video, to perform physical activity through videos of physical exercises to be performed in the house;
- a smartband or smartwatch wearable by the user equipped with heart rate detection (HR sensor) and Bluetooth connection for communication with other IoT devices;
- beacons for indoor localization of the user wearable device or robot;
- a smartphone for the caregiver to receive any notifications from the robotic system for dangerous situations or requests for intervention.

The robot will need to incorporate a variety of user support services:

- The robot shall be able to offer a variety of entertainment services including games, videos (e.g., documentaries, video recipes, physical exercises to be performed at home), and music via the tablet with which the robot will be equipped. These types of entertainment will be tailored to the preferences of each user.

- Of fundamental importance is the reminder function that will have to be established according to the times of a user's habits or medical prescriptions, if it is about taking medicines.

- In order to make the robot interact with the user, AI algorithms will be implemented for user recognition. The robot will then be able to navigate around the house looking for the user by taking advantage of the built-in video cameras to recognize and locate the user's presence.

- If the robot has low batteries it must locate the charging base and connect to it to recharge itself.

- The robot interacts with the user through tablet and voice suggesting activities to be performed daily such as taking medicine or asking if a certain activity has been performed to verify the correct execution of daily habits. In the case of a negative answer, the robot will alert the robotic system of the anomaly to eventually contact the caregiver or a family member in case of need.

- The robot's built-in cameras will detect the person's emotions and activities.

- The robot must be able to communicate via Bluetooth with other devices: - the user's wearable device to detect the user's pose, positioning, and heart rate and pass this information to the robotic system; - beacons to locate the robot inside the house.

The user-wearable smartband or smartwatch will have the following functionality:

- Heart rate detection by HR sensor
- detection of the user's position inside the house by calculating the distance of the device with respect to the beacons through Bluetooth connection
- detection of the user's pose (standing, sitting, working on computer, walking, running, climbing stairs) by exploiting the motion sensors (accelerometer, gyroscope)
- Bluetooth communication to pass on-demand information to the robot, derived from the previously described functionalities.

Beacons are devices that transmit low-power radio waves by exploiting Bluetooth technology to identify the presence of devices. Therefore, two devices will be placed inside the house to identify two rooms most used by the robot for user interaction and monitoring. Two possible scenarios need to be distinguished:

- The robot identifies the distance between itself and the two beacons, identifying its position inside one of the two rooms based on the closest detected beacon.
- The robot identifies the user's location within one of the two rooms by making a location request to the user's wearable device, which will detect the nearest beacon to recognize the location within one of the two rooms.

The smartphone that will be provided to the caregiver will be the means of communication between the robotic system and the caregiver for any requests for help or to notify any anomaly found by the robotic system during the monitoring of the performance of daily activities of the user.

3.2 The Proposed Architecture

The analysis of the functional requirements described in 4.2 was the first step for the formalization and implementation of the robotic system I present in this section, based primarily on user personalization and profiling.

Other intelligent robotic systems, such as the project "Mobiserv" [54], also focus on health monitoring (using wearable sensors), safety, and nutritional support. In [54], for example, the robot can remind the user to drink or eat something, do physical exercises or a puzzle game, and monitors through sensors the gas, water, windows and doors by checking if they are open. These robot behaviors and scenarios are adapted through customization of the system according to personal needs, as in our project. Similarly, in the project "SocialRobot" [64], the Authors developed a mobile robotic virtual social assistance platform that meets the personal needs of the elderly while maintaining the users' self-esteem during daily routines. This platform adapts to daily events during human-robot interactions through emotion recognition and empathic interaction.

Testing in real-world contexts for home-based assistive care of the elderly within their own apartments is the innovation that differentiates us from the other projects already discussed in section 2.1. In the other projects, they tested their platforms in nursing homes or assisted living facilities, i.e., in controlled environments that often consist of IoT sensors that are invasive to people.

An additional innovative aspect is user adaptivity and profiling i.e. an adaptive approach that is easily customizable. This additional aspect is the basis of our work and the framework I present in this section. An example is in [35], where an adaptive approach is proposed that can be customized even by non-expert users such as caregivers. Through a questionnaire, the caregiver customizes the care protocol. The answers obtained are then used for the automatic generation of the daily schedule.

What I present is the schema of the framework from the perspective of the planner, a fundamental component for customizing the behaviors of the robotic system according to the user's needs. Next, I instead present the schema of the framework from the point of view of the communication between the various modules that offer services to the user.

3.2.1 The proposed framework from planner perspective

The proposed framework for the generation and the execution of personalized assistive plans for home patients affected by neurological disorders is composed of different modules organized according to a layered architecture depicted in Figure 3.1. The aim of the design is to decouple low-level functions for managing devices, for data elaboration, and for basic robotics behaviors, from high-level functions adopted for reasoning on the assistive plans. The lower layer is the Daily Assistive Workflow Generator (DAWG) [19] [18], a middleware responsible for the generation of a personalized set of assistive tasks, named a Daily Assistive Workflow (DAW), and for its reconfiguration when changes are detected by the Smart Environment. A DAW represents the flow of the activities that the robot, or even other devices, must perform to monitor the patient, and to interact with him/her. The DAWG is composed of different modules: the first processes the daily routine of the patient extracting the activities to be monitored, that are considered as goals to be fulfilled. Starting from the set of goals (according to the goal model), the encoded user profile, and the high-level observations deriving from the interaction of sensors with the environment and the elaboration of such data, the DAWG selects the assistive actions able to fulfill the goals, named an Abstract Assistive Actions. They are represented as parametric actions that have to be configured according to the patient's cognitive and personality profile. The configuration consists of selecting a specific action to execute, named a Daily Assistive Action (DAA) that is a concrete instance of an abstract assistive action. The separation between abstract and concrete actions is adopted to manage personalization and adaptation of the DAW, decoupling the general description of a certain action from its actual implementation concerning the way it is performed. In details, an Abstract Assistive Action specifies the high-level interface of a certain functionality, including its input parameters, preconditions, and possible outputs. Conversely, a Daily Assistive Action represents the actual implementation of the action executed by using the suitable Sensors and Actuators nodes provided by the Smart Environment. For each Abstract Action, a list of several Daily Assistive Actions may realize the

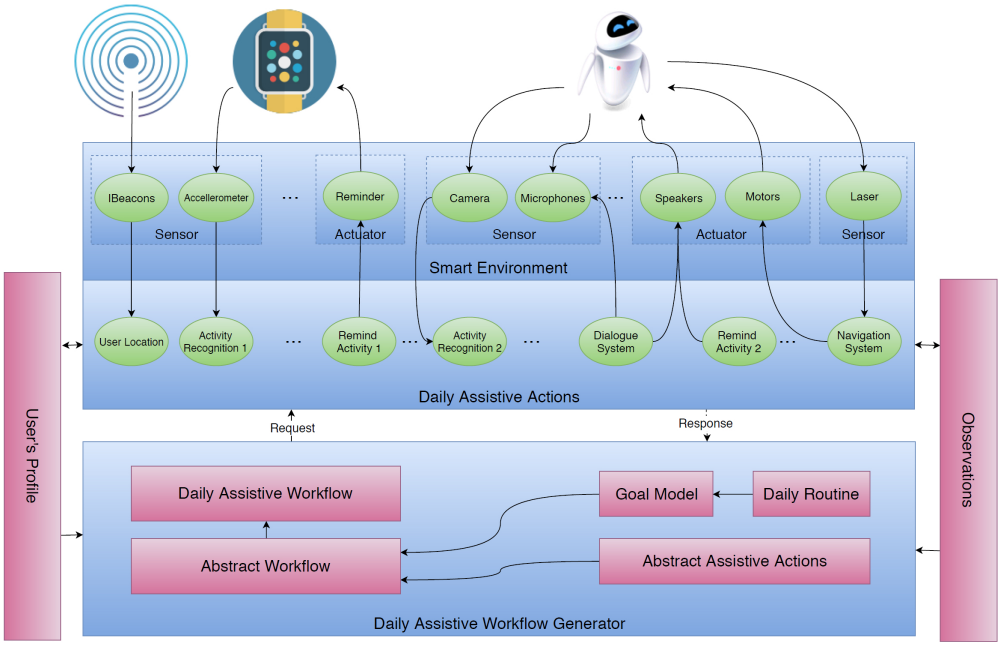

Figure 3.1: The proposed framework from the perspective of the planner

same functionality. Moreover, effective planning of the activities will also have to consider contingent situations that may affect the patient's particular conditions, and so his/her possible habits, e.g. an activity involving the control that the patient has taken his medication may no longer be necessary if the patient had an unexpected medical necessity. The middle layer is composed of DAAs. The goal of such actions is to effectively provide either different algorithms for analyzing input data to monitor the user state and behaviors (i.e., using different input data and modalities to obtain such information), and so updating the observations and the user profile, but also to implement different navigation and interaction strategies to be used by the robot.

This approach is in the direction of integrating robot functionalities (DAAs) as services that can be requested for the seamless integration of robots, as well as other IoT devices, into a web or cloud computing environment [37], or Robot as a Service (RaaS) [9]. In this Service Oriented Architecture view of the assistive domain, RaaS are endowed with such functionalities, or services, to control their behavior as well as to provide meaningful observations from the input data [66].

Moreover, a robot could use different services to provide the same functionality, and a service could be shared and used by different robots. Some of these services will be requested by the execution of the Daily Assistive Workflow, while others are autonomously running or activated by events. To obtain a better adaptation to the user, the project proposes to equip the user cognitive profile with a psychological personality profile, and to adapt the robotic behavior not only with respect to the choice of the single activity to be undertaken (selected by the DAW), but also with respect to the way in which the same activity is performed. Indeed, in order to be effectively deployed, also the robot should be able to regulate its social interaction parameters (e.g., the interaction distances, proxemics, the speed of movements, and the same modality of interaction) based on personality factors as well as of the cognitive state of the user. Hence, the user profiling plays a fundamental role both to generate a DAW tailored for each patient, but also to modulate the execution of Daily Assistive Actions. In fact, according to the personality of a patient, some actions can be performed with a different interaction modality, such as direct interaction with the robot if the user is in a state of inactivity and calm, or remote interaction with the robot staying at a certain distance, if the user is in a state of agitation. The upper layer is represented by the Smart Environment composed of sensors and actuators that play the twofold role of gathering information on the patient's state, and of performing assistive actions. Low-level functionalities make direct use of sensors and actuators installed into the Smart Environment, which are respectively managed by the DAA. Figure 3.1 shows how low-level nodes are combined to compose high-level functionalities.

3.2.2 Daily Assistive Workflow Generator

The Daily Assistive Workflow Generator (DAWG) is the component responsible for the generation of monitoring assistive plans. In order to perform this task, the DAWG takes into account the user's profile described along with the daily routine, the current observations and the entire set of Abstract Assistive Actions, and Daily Assistive Actions. The daily routine is represented by a set of activities that

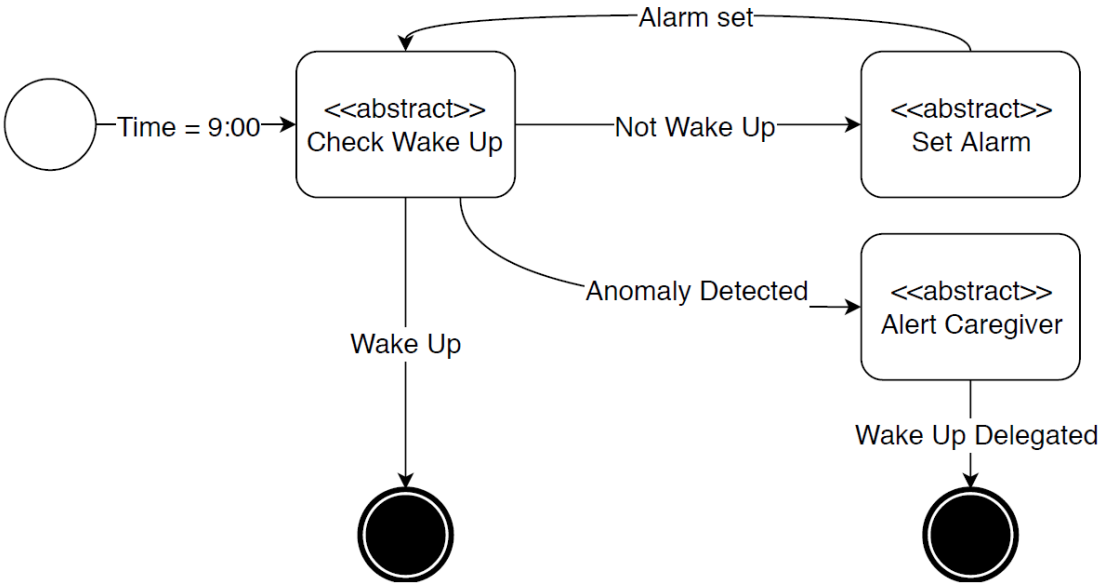

Figure 3.2: Example of an abstract workflow generated for the *Wake-Up* activity

the user has to perform throughout the day, each of them labeled with a time constraint. The daily routine is then encoded into a set of goals, one for each activity, that the system has to achieve with respect to the time constraints. The workflow generation process consists of two main steps. The first step is responsible for the generation of the abstract daily assistive workflow, representing the set of actions to be scheduled to monitor the daily routine of the patient, organized as a set of goals to be fulfilled. Figure 3.2 shows an example of an abstract workflow for the *Wake-Up* activity, representing the set of abstract actions necessary to monitor the *Wake-Up* activity. Each Abstract Assistive Action has to be instantiated by a concrete Daily Assistive Action in order to be executed. Therefore, an abstract workflow represents a high-level template for the sequence of actions required to achieve a certain goal. The second step is responsible for the instantiation of a specific Abstract Assistive Workflow. This process starts as soon as a certain time constraint triggers a new goal. For instance, with respect to the workflow shown in Figure 3.2, the instantiation process will be activated at 9:00 am by triggering the corresponding goal Wake Up at that time. When a goal is triggered, the system retrieves the abstract workflow associated with the current activity to be

monitored and turns it into a concrete workflow. The instantiation process follows the structure of the corresponding abstract workflow. Its general structure is represented as a graph $G(V; E)$ in which each vertex $v \in V$ represents an abstract action to be instantiated and each edge $e \in E$ represents a transition labeled with a condition. Starting from the first vertex, each abstract action is instantiated by selecting the most suitable one among the available concrete Daily Assistive Actions. For instance, in Figure 3.2, the abstract assistive action *Check Wake Up* may be implemented by different concrete actions offering the same functionality with different modalities. If we consider the environmental setting depicted in Figure 3.1, *Check Wake Up* can be actually realized by the Activity Recognition module provided by the Smartwatch, as well as by the Robot via Camera. Moreover, even the Robot's Dialogue System can be suitable for this task. The main characteristics we consider to differentiate a concrete action from each other are its reliability and the interaction modality. These non-functional parameters are then matched against the user profile to determine a ranking over possible concrete implementations to select the one that represents the best trade-off between user needs and reliability. Once an abstract action is instantiated, the selected concrete implementation can be executed by the corresponding device, e.g., the robot. In addition, the concrete action can be executed in different modes (e.g., interaction modes), i.e. with different values of some nonfunctional parameters. Here, the execution of a certain action produces as output new observations deriving from sensors installed into the environment. These observations are used to determine whether a certain state is reached. Hence, the system is able to determine the transition to the next vertex in the graph after the execution of each concrete assistive action. When a final condition is reached, the workflow execution is completed and the system waits until a new goal is triggered.

3.2.3 The software architecture

The social multifunctional robot system, presented in this work, is composed by an high-level layer, the WorkFlow Manager, and a low-level layer composed by

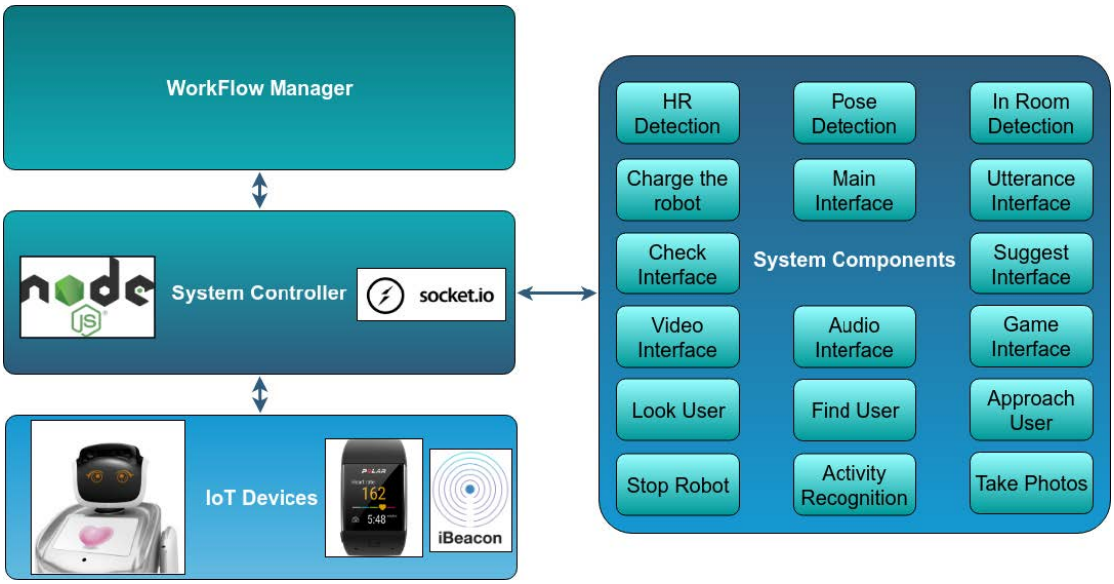

Figure 3.3: The software architecture

the robotic behaviors. In [17], we laid down the general guidelines underneath the architecture of the Workflow Manager, consisting the high-level layer, and emphasized the intent of decoupling low-level functions (e.g. managing devices, data elaboration, and basic robotics behaviors) from high-level functions, like reasoning on the assistive plans and generate a personalized program, tailored specifically for elderly people affected by dementia. In this work, I focused on the lowlevel functions for managing devices, in particular on the robot behaviors. Much like the Robot-Era [6] and SocialRobot [63] projects, I deployed a RaaS architecture.

An architecture similar to ours, with at the center of the daily schedule an engine that generates daily workflows, is proposed in [64]. Underlying that work, the Authors present a Workflow Engine that interprets service requests through XML messages. These services are then executed through ROS (Robot Operating System) services. Another approach more widely used is the use of the cloud through the Robot as a Service (RaaS) architecture to provide greater scalability, elasticity and computational power as in [6].

In particular, I want to improve the deployment of the social assistive robot's behaviors relying on the use of a series of microservices, to allow greater adaptability of the system. The aim is to create a scalable and highly modular architecture,

obtained by dividing all functionalities required from the robot into a series of basic and primitive cohesive functionalities - that we call microservices. Each microservice can constitute itself a full system component or it can be combined in cascade with other basic microservices, to provide a more complex functionality as a higher-level component. Moreover, these smaller microservices rely on different programming languages, and may have disjoint dependencies, container and responsibilities. The strong modularity and scalability of the proposed architecture ensure ease of maintenance, and makes it easier to expand it whenever necessary, without having to modify or rewrite other modules. In [22], the authors introduced the most important features of the microservices compared to a monolithic approach: the microservices are independent and directly testable in isolation with respect to the whole system; they foster continuous integration for an easy maintenance; changing a microservice does not require a reboot of the system; they have their own container with their own configuration of the deployment environment; scaling is simple and it does not require a duplication of all components; the only constraint is the technology for the communications between the microservices. Such a service oriented architecture (SOA) integrates the robot functionalities into a Robot as a service (Raas) unit [9] [6], which is a cloud computing unit. This allows to share these services among different robots with different strategies depending on the observations and the user's profile.

3.2.4 IoT Devices

The hardware infrastructure used for this project comprises a Sanbot Elf robot (Figure 3.4), a service robot, developed by Qihan Technology, equipped with infrared sensors, omnidirectional locomotion, a full HD touchscreen (which is basically an Android tablet) and an RGB-D camera; a Polar Android M-600 smartwatch that mounts accelerometer, gyroscope, optical heart rate measurement with six LEDs; iBeacons, used for room labeling, capable of transmitting a signal using Bluetooth Low Energy (BLE) technology (the strength of the RSSI - Receive Signal Strength Indicator - was used to define proximity relations); an Android

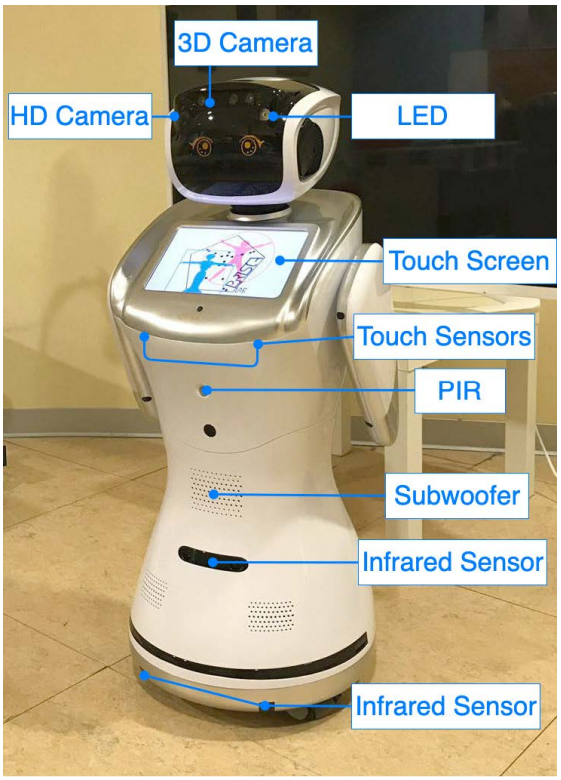

Figure 3.4: Sanbot Elf robot and its hardware components

smartphone. The center of most of the computations, especially those regarding operations requiring the robot to move, is the tablet mounted on the robot. The Android APIs allow to effectively exploit all the functionalities featured by the robot.

3.2.5 Robot Server

The main server is composed by four different modules: the WorkFlow Manager [17], the System Controller, the System Components, the IoT Devices (Figure 3.3). All these modules are executed on a computational unit connected in a private network with the robot. Indeed, during the final experimentation phase, the entire system will be deployed in a real-world scenario with elderly affected with Alzheimer's disease. In order to comply to privacy and security constraints, the robotic platform will rely on a private Wi-Fi network, but without any Internet connection. The System Controller is a server implemented using Node.js", which

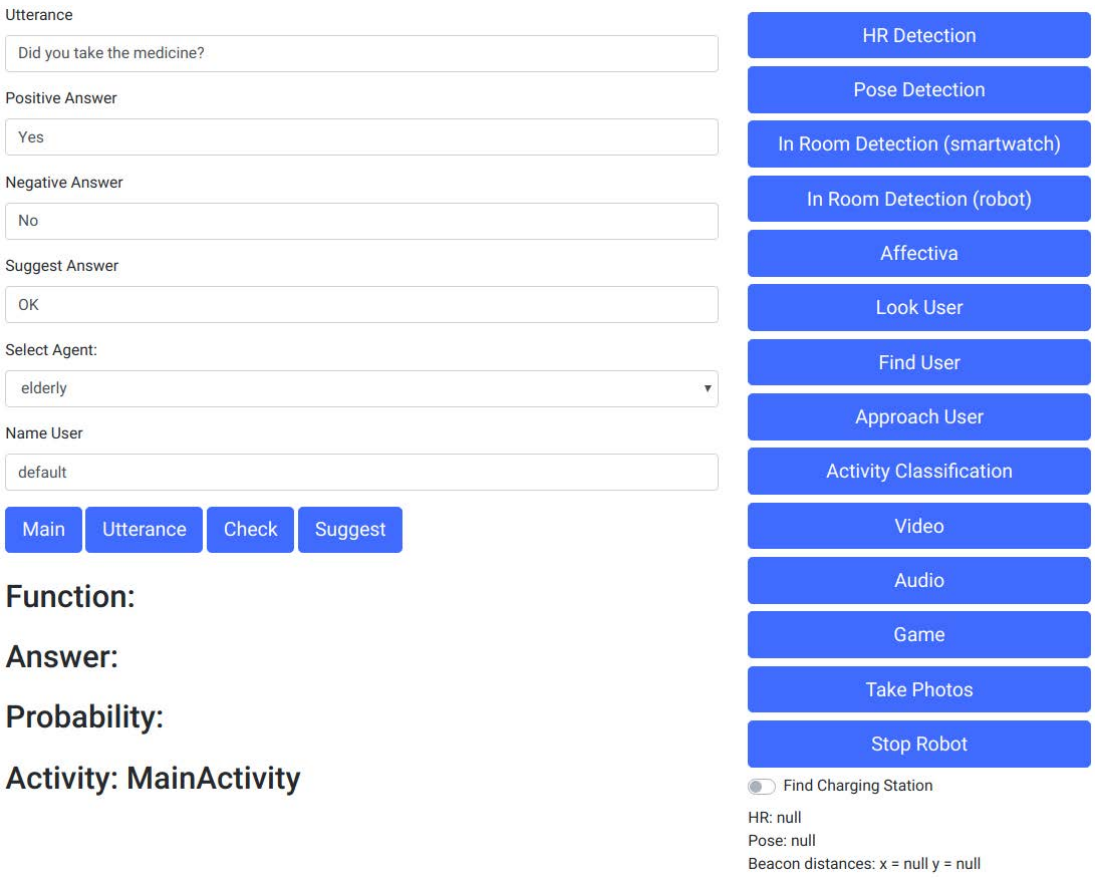

Figure 3.5: Control Interface for testing the System Components

allows to handle the I/O functionalities asynchronously. It handles the communications between all the modules of the framework with the event driven architecture based on the "Socket.IO" library providing real-time communication between the modules. The communications channels are implemented using web socket, while the message format is defined using JSON. Thanks to the communications protocol employed, each component can contact the others to request/response to a particular need going through the System Controller. I have also developed a Control Interface (Figure 3.5) for testing the System Components implemented with which I checked the correct functioning of all the services. The Control Interface is a web page built using the "Express" library that is a minimal and flexible "Node.js" web application framework. Assistive plans for the robot (but also including actions to be performed on other IoT devices) are handled by using a module responsible for their generation and for their dynamical reconfiguration

[19]. The module generates a workflow of services, according to a service oriented approach, obtained by automatically combining the Microservices that can be executed by the robot or other available devices.

3.2.6 Implemented Services

The implemented services are divided into services for monitoring, for navigation and for interaction:

- Monitoring Services
  - HR Detection
  - Pose Detection
  - In Room Detection (wearable-based)
  - In Room Detection (robot-used)
  - Emotion Recognition
  - Activity Recognition
- Navigation Services
  - Find Charging Station
  - Find User
  - Look User
  - Approach User
- Interaction Services
  - Main Interface
  - Utterance Interface
  - Check Interface
  - Suggest Interface
  - Video Interface

- Audio Interface
- Take Photos

Communication between services is based on event-driven communication. The server, which handles the communication via the "Socket.IO" library, then broadcasts an event thrown by a service to reach the right recipient based on the event thrown. I will explore the following services in more detail in the chapter 4.

Chapter 4

Services

In this chapter, I first consider an overview of the software analysis in section 4.1 and provide a more detailed description of some of the implemented system components in the section 4.2 and 4.3. Specifically, these components are divided into three main groups, which perform different types of functionality, respectively: A) Monitoring Services in section 4.4, which include activity recognition via a wearable device or via camera using pose/skeletal recognition, emotion recognition, and disengagement; B) Navigation Services in section 4.5, for user search and approach; and C) Interaction Services in section 4.6, for speech recognition and synthesis using multimodal user interaction. In section 4.7 I discuss about the communication between the services and in section 4.8 I show the tests conducted on the response times of person finder services within the simulated home environment.

4.1 Software Analysis Overview

Below is a detailed description of the functionality required of the robotic system following the objectives set.

4.1.1 Product outlook

The robotic system is a completely autonomous system that will not have to consider the intervention of any operator, managing the user's data within its own local network without any external connection to ensure the privacy of personal data.

4.1.2 Product Functionality

The robotic system must:

- Manage patient wearable device information:
  - HR Detection: Heartbeat detection.
  - In Room Detection by wearable: Detection of the patient's position relative to the beacon references
  - Pose Detection: Detection of patient pose
- Provide entertainment and physical/mental stimulation services according to user customizations via the robot:
  - Game: Entertainment game
  - Audio: Music playlist with automatic playback or choice of tracks
  - Video Entertainment/Physical Exercises: Videos of physical exercises to do at home, documentaries, cooking, traveling or other
- Manage Robot Information:
  - Look User: Detecting user presence.
  - Find user: Navigate through the home space to find the user.
  - Approach: Approach of the user to interact with the user
  - Check/Suggest: Text-based interaction for assertions, verification, and suggestions by the robotic system for the benefit of the patient

- Find Charging Station: Charging the robot via the charging station Affectiva: Emotion detection.
- In Room Detection by robot: Detection of the robot's position relative to beacon references
- Activity Recognition: Detection of activities performed inside the house

4.1.3 User characteristics

The robotic system is aimed at users aged between 50 and 80 years without particular computer skills.

4.1.4 General Constraints

The video information should not be available to the operators for privacy reasons so it will not be saved. The robotic system devices will not have access to external networks such as the Internet. Services must be customized according to user preferences. It will be possible to save only the success/failure information of the services offered through the robotic system.

4.1.5 Assumptions and dependencies

The robotic system will have to be equipped with a router for internal network management and interconnection between the various IoT devices. Data processing and planning of the daily behaviors of the robot will be performed on a small personal computer running Ubuntu 18.04 operating system, due to the reduced computational power of the robot, which will act as a server for the robotic system.

4.1.6 Specific Requirements

User Interface

Interactions with the robotic system will be through the robot's tablet with a very simple interface with text and buttons that will appear according to the schedule

of daily behaviors that the robot will have to follow. The user will only be able to interact with the robot.

Hardware interface

The communication between the various IoT devices will take place via Wi-Fi in a local network managed by a router and via Bluetooth for communication between the beacons, the wearable device (smartband/smartwatch) from the patient and the robot.

Software interface

Software communication between the various separately developed components will be via websocket.

4.2 Functional Requirements

HR Detection Requirement

Introduction Allows detection of the heartbeat of the user wearing a smartband or smartwatch.

Input Name of the required service.

Processing The IoT device via the heartbeat detection sensor records a 15 second window of the readings.

Output Returns the average heart rate over a 15 second window or null due to a timeout or other exception.

In Room Detection by the wearable Requirement

Introduction Allows to detect the location of the user wearing a smartband or smartwatch.

Input Name of the requested service and user ID.

Processing The IoT device connects to two beacons installed in two different rooms: living room and kitchen. The distance to the beacons is calculated based on the Bluetooth signal strength.

Output Returns the location of the user, living room or kitchen or null in case of no connection to the beacons.

Pose Detection Requirement

Introduction Allows to detect the pose of the user wearing a smartband or smartwatch. Note that detected activities can be combined with detected poses, but they are two separate classifiers.

Input Name of the required service.

Processing The IoT device records 512 samples of the accelerometer data. Processing of the data for pose classification is done on the server.

Output Returns null in case of connection problem or one of the following poses: lying, sitting, standing, walking, watching TV, working on computer, ironing.

Game Requirement

Introduction It allows to entertain the user through some games like a card game or tic tac toe.

Input Name of the service required and type of game.

Processing The robot's tablet will display the chosen game on the screen.

Output Returns stopped if the game was finished by the user before its completion, finished if the game was completed or went into timeout, null in case of connection problems.

Audio Requirement

Introduction Entertains the user by playing a music playlist.

Input Name of the required service and type of playback, random if it should play the tracks of the playlist randomly, select if the user should be able to select the tracks to be played.

Processing The tablet of the robot will display the music playlist on the screen playing the songs of the playlist.

Output Returns stopped if the user has stopped the music playback or has timed out, finished if the user has listened to the whole playlist, null in case of connection problems.

Video Entertainment/Physical Excercises Requirement

Introduction It allows to entertain the user playing videos among which a list of documentaries, cooking videos, travel videos and physical exercises to do at home to keep fit and active.

Input Name of the requested service and type of playback, entertainment if it has to play videos of documentaries, travel, cooking, exercise if it has to play videos of physical exercises to do at home.

Processing The robot's tablet will display and play the requested videos based on the type. A set of videos for each type will be prepared and will be chosen randomly from the required type.

Output Returns stopped if the user stopped the playback or went into time-out, finished if the whole video was played, null in case of connection problems.

Look User Requirement

Introduction It allows to detect the presence of the user in the vicinity of the robot that remains fixed on its position.

Input Name of the requested service.

Processing The robot searches for a person by remaining in its position, performing four rotations to get a total view of the surroundings. It is not possible to discriminate the person by face recognition if the distance between the robot and the user is not adequate. To perform face recognition of the person, the robot must stand at a maximum distance of 1.5 m and the person must be in the field of view of the robot's front camera.

Output Returns found if the user is in the field of view of the robot's front camera, null in case of connection problems.

Find User Requirement

Introduction Detect the presence of the user in the vicinity of the robot while moving around in the home environment.

Input Name of the required service.

Processing The robot searches for a person by staying in its position, making four rotations to get a full view of the surroundings. If the person is not around the robot during the four rotations, the robot will start moving randomly in the home environment to find the person. It is not possible to discriminate the person by face recognition if the distance between the robot and the user is not adequate. To perform face recognition of the person, the robot must stand at a maximum

distance of 1.5 m and the person must be in the field of view of the robot's front camera.

Output Returns found if the user is in the field of view of the robot's front camera, null in case of connection problems.

Approach Requirement

Introduction Allows the robot to approach the designated person.

Input Name of the requested service, identifier of the person to be approached by the robot.

Processing The robot, having found the person in its vicinity, approaches it in order to recognize the face.

Output Returns found if the user is in the field of view of the frontal camera of the robot and is the person corresponding to the identifier given as input, otherwise null, even in case of connection problems.

Check Requirement

Introduction Allows the display on the robot's tablet of a textual question with two buttons below to answer positively or negatively to the question.

Input Name of the requested service, text for the question, text for the positive answer, text for the negative answer.

Processing A textual question with two buttons underneath for a positive or negative answer to the question is displayed on the robot's tablet.

Output Returns the answer selected by the user via the tablet, null in case of connection problems.

Suggest Requirement

Introduction Allows a suggestion to be displayed on the robot's tablet with an underlying button to accept the suggestion.

Input Name of the required service, text for the suggestion, text for the button below the suggestion.

Processing A hint with a button underneath to accept the hint is displayed on the robot's tablet.

Output Returns the answer selected by the user via the tablet, null in case of connection problems.

Find Charging Station Requirement

Introduction Enables or disables finding the charging station to charge the robot.

Input Name of the required service, Boolean value to enable or disable the charging base search.

Processing The robot moves around the home environment searching for the charging base recognized by infrared sensors.

Output Returns in_charge if the robot is charging, not_in_charge if the robot is not charging or in case of timeout, null in case of connection problems.

Affectiva Requirement

Introduction Allows you to detect the emotion felt by the user via the robot's front-facing camera.

Input Name of the service requested, person identifier.

Processing The person is recognized and the facial expression is analyzed to derive the emotion felt.

Output Returns null in case of connection problems or one of the following emotions: joy, surprise, contempt, disgust, sadness, anger.

In Room Detection by the robot Requirement

Introduction Detect the location of the robot.

Input Name of the required service.

Processing The robot connects to two beacons installed in two different rooms: living room and kitchen. The distance to the beacons is calculated based on the strength of the Bluetooth signal.

Output Returns the location of the robot, living room or kitchen or null in case of no connection to the beacons.

Activity Recognition Requirement

Introduction Allows to recognize the activity that the user is performing daily in the home environment. Note that detected activities can be combined with detected poses, but they are two separate classifiers.

Input Name of the required service.

Processing The robot analyzes the user's movements by recognizing the user's sequence of poses, represented by a set of points that identify the joints of the human skeleton.

Output Returns null in case of connection problems or the activity that the user is performing, among which we have: using the PC, calling the phone, watching TV, preparing coffee, ironing clothes, talking on the couch.

4.3 Performance Requirements

Interaction times must be within the right limits for better acceptability of the robotic system.

4.3.1 Design Constraints

The local network of the robotic system must not have any external access to other networks such as the Internet in order to respect the privacy of the patient adopting this robotic system in their home.

4.3.2 Software System Attributes - Security

The information processed by the robotic system is strictly confidential. Therefore, no audio video data will be recorded in the robotic system. It will be possible to process, at the end of the in-home experimentation period, only the textual data of interaction between the robot and the patient to record the success or failure of the interactions to assist the person in the daily planning of household activities.

4.4 Monitoring Services

Monitoring services are developed to estimate and recognize the current state of the user, and of Activities of Daily Living (ADLs) and of Instrumental Activities of Daily Living (IADL) he/she is occupied performing.

- HR Detection: a Bluetooth Socket connection is established between the robot and the smartwatch. The smartwatch detects the heart rate of the user for approximately 15 seconds, after which the mean of the readings is returned.

- Pose Detection: the robot and the smartwatch communicate via Bluetooth Socket. The smartwatch gathers 512 accelerometer and gyroscope data samples that are returned to the robot. These information are sent to a Python module, using the Socket.IO-based communication system, where a Deep

Neural Network recognizes and outputs the activity performed, which is the one with higher prediction probability among the following activities : lying, sitting, standing, walking, watching TV, computer work, and ironing. The Pose Recognition module consists of two LSTM layers that perform temporal features extraction, and a fully connected layer, which outputs the classification probabilities of the input to belong to each of the classes (activities).

- In Room Detection (wearable-based): the application running on the robot establishes a Bluetooth Socket communication with the Android application installed on the smartwatch and requests the location of the user in the house. The smartwatch, on its end, fetches the distances from the reachable beacons placed in each of the rooms considered for the experiments. Then, given the distances from the beacons to the smartwatch, the label of the room with the smaller corresponding distance is returned to the robot.

- In Room Detection (robot-based): following the same procedure of the wearable-based In Room Detection, the robot retrieves the distances from the beacon placed in reachable rooms, which these devices transmit in broadcast. The label associated to the room with the smaller distance is returned in this case as well. If this service is called after the robot has approached the user, it provides the location of the user. Moreover, since the charging station must be at a distance ranging from 0 to 5 meters from the robot in order to be detected, this functionality can be employed to find the correct room towards which the robot should navigate.

- Emotion Recognition: this service relies on the Affectiva SDK [52]. The robot records a video of the person and transmits it to the Affectiva module, which returns the emotion detected (joy, surprise, contempt, disgust, sadness, anger), the engagement and the valence of the person with the highest mean in the whole video.

- Activity Recognition: using the robot's camera, the service records a video

of the user. From the clip obtained, for each frame, the skeleton coordinates are extracted, using TensorFlow's pose estimation model, PoseNet [32] [60] [59], which estimates where the key body joints are. The activity recognition model that will rely on the skeleton joints, is implemented according to [28]. The Deep Neural Network employed consists of two LSTM layers and a fully connected with softmax activation function that outputs the probabilities of each class to belong to each of activity considered. The model has been trained on a dataset built using data gathered during the experimentations of a previous work [69], that took place in our laboratory. The video footage was shot the camera of the robot. The activity performed, and therefore constituting the dataset, are watching tv, relaxing on couch, ironing, making coffee, working at PC, and talking on the phone.

4.5 Navigation Services

These services provide the robot the fundamental functionalities required to navigate within the environment and to locate the user. The considered low cost devices do not allow to properly manage localization and mapping tasks. Hence, in order to rely on hardware with limited capabilities, a reactive approach has been employed for navigation.

- Find Charging Station: when activated, the service instructs the robot to wander towards the room where the charging station is located (using the robot based In Room Detection), and then wanders in that room to find it, leaning on the infrared sensors it is equipped with.

- Find User: this service provides the wandering functionality, for which the robot wanders in the apartment and avoids obstacles for one minute. In order to find a user, this service has to be combined with the "Look User" service.

- Look User: the robot rotates on itself to look for a person that is in the proximity. The visual field angle of the robot camera is about 100 degrees,

but to obtain a complete scan of the environment, the robot has to rotate four times on itself. PoseNet is used to check if a person is in front of the robot.

- Approach User: the user approach service relies on the evaluation of the user pose from his/her skeleton data. A different approaching direction can be computed according to the approach developed in [27]. When a user is detected, the FaceNet algorithm [77] is deployed to establish if he/she was the person it was looking for.

4.6 Interaction Services

Interaction modalities of the robot (voice interaction and GUI) can be used to suggest and show personalized entertainment activities to the user.

- Main Interface: it is the main window showed on the Android tablet of the user, which is launched initially when the application starts. No input parameters are returned.
- Utterance Interface: an interface with an utterance is shown on the robot tablet to communicate something to the user.
- Check Interface: an interface with a question and two buttons (for positive and negative answers) is shown on the robot tablet (i.e., to check if the user has taken the medicines).
- Suggest Interface: an interface with a suggestion and a button, to check if the user accepts the suggestion, is shown on the robot tablet.
- Video Interface: a video is played on the tablet of the robot. The video is randomly chosen from a list of videos considered relevant to the patient, but it is also possible to specify it with an URL.
- Audio Interface: a list of songs relevant to the patient is showed on the robot tablet; the songs can be played randomly or chosen by the user.

- Game Interface: a chosen game is showed on the robot tablet relevant to the user preferences.

- Take Photos: the robot records a video of the user and saves on the "Robot Server" to extract the face features; that will be used to populate dataset of the feature vectors extracted by the FaceNet to recognize the user.

4.7 Communication

Socket.IO is highly appreciated by Node.js developers because it allows synchronized communication between client and server in real time. The most used applications with Socket.IO are chats on websites but the possibilities that Socket.IO offers go far beyond that. Socket.IO is a library that allows us to simplify the implementation and logic of communication between software and hardware components. One of the possible communication channels used by this library is WebSocket. It allows synchronized bilateral exchange between client and server. WebSocket therefore allows you to create an always open communication channel between client and server. In addition to WebSocket, Socket.IO offers the possibility to use other similar methods of real-time communication, each to best suit each client: WebSocket, Adobe Flash Socket, AJAX Long Polling, AJAX Multipart Streaming, Forever Iframe, JSONP Polling. This library therefore abstracts from the implementation of each of these techniques allowing you to speed up development and focus on high-level functionality. In order to send a message between client and server or vice versa an event must be emitted associated to the data: to do this, Socket.IO uses an event driven architecture. This library makes it very easy to send events to all clients connected to the server and also allows to send from the server in broadcasting a message to all connected clients. Sending messages in broadcasting has therefore allowed to have a bidirectional communication also between clients through the server, based on the event launched during the communication, which opens a communication port between two clients.

The architecture of the proposed robotic system is based on various IoT de-

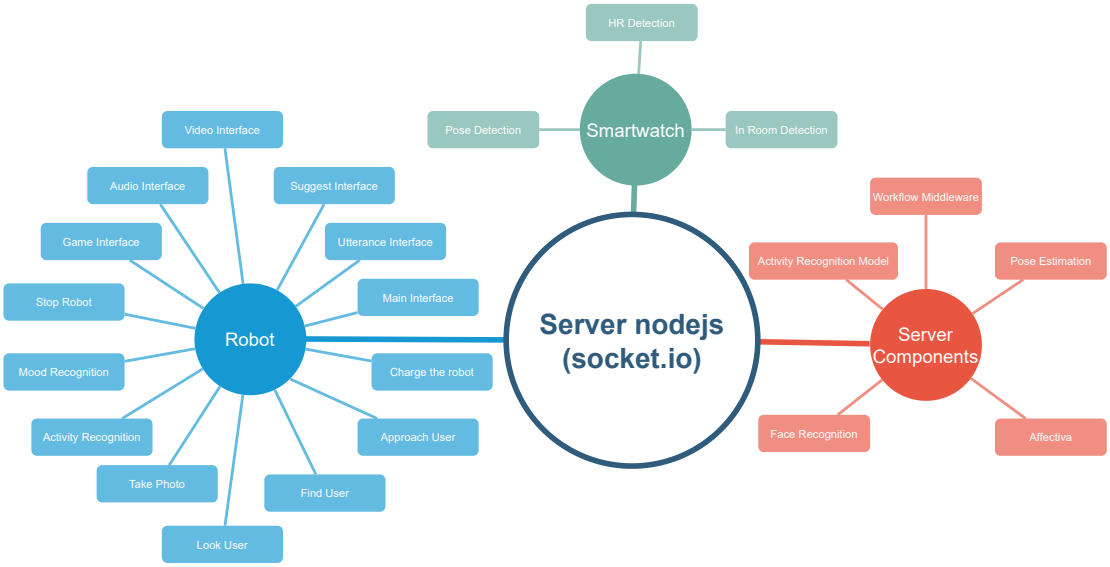

Figure 4.1: A diagram with the principal IoT components that run the different services.

vices. Each of them implements modules that interface with the main server. To understand the complexity of the system I present a diagram that represents the various modules actually implemented in the system. In Figure 4.1 we have "Robot", "Smartwatch" and "Server Components" that communicate with the central node, represented by the "Server nodejs (socket.io)". The central node is the main server script that is executed in Nodejs and exploits the communication through the library Socket.IO. Socket.IO uses a protocol based on events to manage the communication between the server and the clients. In turn, which are nothing more than classes in the Java programming language: Main Interface, Utterance Interface, Suggest Interface, Video Interface, Audio Interface, Game Interface, Stop Robot, Mood Recognition, Activity Recognition, Take Photo, Look User, Find User, Approach User, Charge the robot. Regarding the Smartwatch I have instead implemented the following Java classes: Pose Detection, HR Detection, In Room Detection. The components module that runs on the server, represented by Server Components, consists of the following modules: Workflow Middleware, Pose Estimation, Affectiva, Activity Recognition Model, Face Recognition. Each module of the robotics platform can communicate via Socket.IO by

launching events that are captured by the modules involved. In order to show a possible interaction between the various modules I have represented in the Sequence Diagram the possible scenario in which the robot looks for the user to ask him a question. In the Figure 4.2 we can observe how the Workflow Manager launches the service to approach the user. An intermediate node of the server, which acts as a bridge between the server and the Workflow Manager, receives the request and communicates to the module regarding the robot to launch the service to approach the elderly user. The Approach User service launches the Find User service. Find User in turn contacts Look User to first find out if the person is around the robot. Look User rotates the robot on itself 360 degrees to check if there are people around. If it is not found, it returns the response to Find User which initiates the wandering of the robot. Then it starts Look User again to search around itself, until it finds the person, with a maximum limit of 10 minutes. In the sequence in Figure 4.2 the person is found and the result is returned to Approach User that will make the robot approach the elderly person in order to interact with him. Once close to the possible target user, it launches the Face Recognition service to verify that is the elderly person. In the example, the robot was able to recognize the elderly person and finally the Workflow Manager is notified of the success. The Workflow Manager continues with planning the daily behavior of the robot by launching Utterance Interface, which shows a question to the user. The user can now answer the question by selecting one of the possible answers. In the example, the user answers yes to the answer, which is forwarded to the Workflow Manager, which will then continue planning accordingly.

4.8 Testing

The proposed framework has been introduced and tested in the house of an elderly couple of 80 and 76 years old without any relevant cognitive impairment, with the objective of testing the whole system in a domestic environment. In order to allow the framework to work properly, it is fundamental that the navigation services enable the robot to find and approach the user in reasonable time. For this reason,

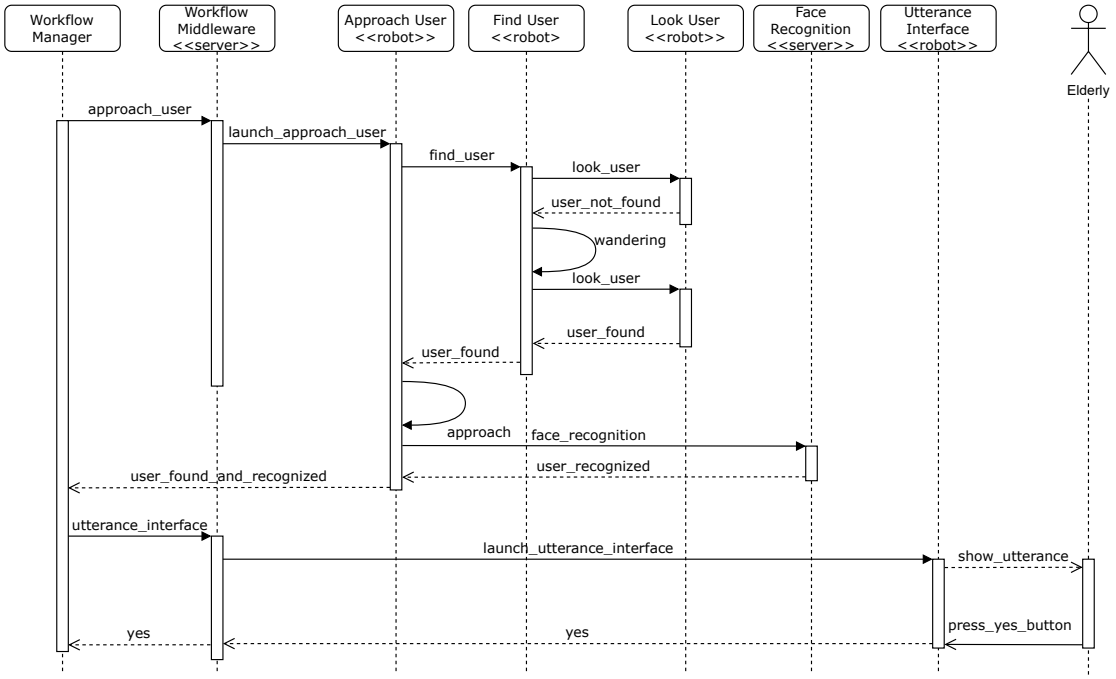

Figure 4.2: A sequence diagram example of the launch of a workflow.

I tested these in the living room of the home of the elderly participants. Specifically, the "Look User" and "Approach User" services have been tested, along with "PoseNet" and "FaceNet" neural network implementations, that are fundamental for their correct behavior. A user has been asked to position himself in the living room. The robot had to find him, recognize and eventually approach him, regardless of the different pose the participant may assume (standing, sitting, or with his knees slightly bend - as if he was sitting in midair), and of the different lighting condition in the room. The "Find User" component, which is the other navigation service provided, launches the "Look User" component, and then instructs the robot to wander in the room for 60 seconds. The modules have been tested separately, verifying how many seconds the functionality would take to successfully perform the task. Three kind of tests were performed without any prior knowledge of the environment. Three experimental settings were individuated as follows:

- In the first set of tests, there are no obstacles between the robot and the person.

Table 4.1: Mean time required (in seconds) for the navigation components.

	Look User	**Find User**	**Approach**
Test 1	$21s \pm 5s$	$66s \pm 14s$	$99s \pm 25s$
Test 2	$19s \pm 8s$	$75s \pm 13s$	$191s \pm 94s$
Test 3	$16s \pm 3s$	$70s \pm 14s$	$115s \pm 59s$
Mean	$19s \pm 5s$	$70s \pm 13s$	$135s \pm 59s$

- In the second, the robot moves in the unknown static environment, with an obstacle between him and the person.
- In the third one, there is an obstacle between the robot and two possible interlocutors. It will have to detect the right person between the two.

The testes have been conducted employing as server an Intel NUC7i7BNH with Intel i7-7567U - 3.50 GHz, 16GB DDR4 RAM and 256 GB SSD.

4.8.1 Results

For the each experimental setting, 10 runs were executed. In the "Look User" service the robot takes a picture and feeds it to PoseNet model. If a human is detected and is in front of the robot, the task is completed, otherwise the robot turns and repeats the process. The "Find User" service provides the wandering behavior and uses the "Look User" service to identify a person. The "Approach User" service, is a system component obtained combining the "Look User" service, the "Find User" service, and the two neural network model employed. For this reason the average time required will be higher than the other three modules tested. However, to prevent the case where the robot continues to search for the user and try to approach him/her for too long, a fixed time-constraint, of 300 seconds, has been introduced. The Approach User reactively calculates the distance from the user, by continuously taking pictures. The results of the testes executed are shown in Table 4.1.

4.8.2 Discussion

The two navigation services aiming at looking and approaching towards a user have been tested. The results show that the proposed services perform the intended task in an adequate and effective way. Both the two services and the neural networks employed executed the tasks in reasonable times. In particular, the "Approach User" service, accomplished in executing the task in 90% of the runs, failing to approach the user in one of the 10 executions. Despite the different configurations of the three tests, the results are very similar. Except for the second approaching test, where the average time is higher than the others. Most likely this result is sometimes due to the difficulty of adjusting the trajectory to avoid obstacles. The testing methodology, followed in this work, aims at evaluating the behavior of the considered modules separately. Therefore, they have not been tested as part of a series of microservices, combined to obtain a workflow handling a specific aspect or situation in the daily routine of the elder. Moreover, the tests have been conducted in a controlled environment, with the users assuming predefined poses. Further evaluations, deploying the whole system proposed, will be produced in new the experimental trials with 40 elderly patients affected with Alzheimer's disease. The services are overall considered to be satisfactory, and we are confident to obtain a positive outcome as well in uncontrolled domestic environment deployment.

Chapter 5

ADL Recognition

Human Activity Recognition (HAR) is a related research field especially in the field of Human-Computer Interaction (HCI) that aims to identify the actions, activities or gestures performed by a human and also aims to predict the possible targets of the action by observing it through sensors. In this project I have focused mainly on the recognition of daily activities carried out by people in their homes, in particular ADLs (Activities of Daily Living) [25]. Activities of daily living are the activities that an adult individual performs independently and without the need for assistance to survive and care for themselves. ADLs are summarized in medicine with the acronym DEATH, which stands for "Dressing Eating Ambulating Toileting Hygiene". ADLs are then distinguished into:

- **Dressing.** Ability to select and wear appropriate clothing
- **Feeding.** Ability to feed oneself in a self-sufficient manner
- **Ambulation.** Ability to move from one position to another and walk independently
- **Continence management.** Ability to properly use the bathroom
- **Personal hygiene.** Washing, bathing or showering, oral, hair and nail hygiene.

In addition to ADLs, Instrumental ADLs (IADLs), activities that are not essential for survival, but allow people to live independently within a community, are also often considered. I focused in particular on iADLs (Instrumental Activities of Daily Living).

IADLs can be grouped into the following types:

- **Company and social support.** A fundamental parameter that evaluates the human resources to support the person in order to improve his lifestyle.
- **Transportation and spending.** Ability to move or obtain groceries and medications without help.
- **Preparing meals**. Ability to plan and prepare various meals, including grocery shopping and storing food properly
- **Cleaning and keeping the house in order**. Ability to clean, tidy, throw away trash, do laundry, and rearrange clothes
- **Medication management**. Ability to obtain prescriptions and medications and to take treatments on schedule and in the correct dosages
- **Communicating with others**. Ability to use communication tools such as the telephone and generally the ability to make the home hospitable and welcoming to visitors
- **Management of finances**. Ability to manage checking account, payments, and expenses.

In this chapter, I introduce the approaches used during experimentation for activity recognition based on a deep neural network described in section 5.1. The proposed models were first trained on the public CAD-60 dataset in section 5.2. I then described a deep model, in section 5.3, trained on a dataset recorded in our lab during experiments with elderly volunteers. An approach used as part of a collaboration with the Sheffield Hallam University to recognize the gestures of autistic children mimicking the gestures of a robot is described in section 5.4.

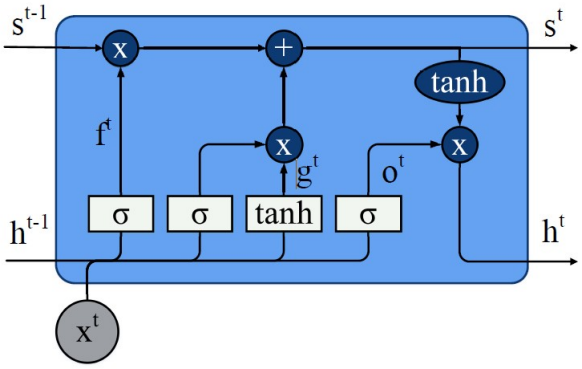

Figure 5.1: A diagram of one LSTM

5.1 The Proposed Approach

The proposed model aims to explore the combination of CNN for representation learning and of LSTM for temporal dependencies learning, that is proposed in applications that concern spatio-temporal classification, like in [21] for video description, and in [58] for activity recognition from wearable devices data.

While the basic RNNs suffer the vanishing/exploding gradient problem [4], the LSTM [38] can handle this problem and learn long-term dependencies. The LSTM can be seen as a block with an internal recurrence in addition to the outer recurrence of the RNN. Every LSTM block is a system of gating units: the state unit s^t (Equation 5.4), the forget gate unit f^t (Equation 5.1), the external input gate unit g^t (Equation 5.2) and the output gate unit o_t (Equation 5.3). An LSTM block receives in input the input vector x^t, the hidden vector h^{t-1} and the state vector s^{t-1} and returns in output the state vector s^t (Equation 5.4) and the hidden vector h^t (Equation 5.5) (see Figure 5.1). The gate and output units are computed using the following equations:

$$f^t = \delta(b^f + U^f x^t + W^f h^{t-1}), \tag{5.1}$$

$$g^t = \tanh(b^g + U^g x^t + W^g h^{t-1}), \tag{5.2}$$

$$o^t = \delta(b^o + U^o x^t + W^o h^{t-1}), \tag{5.3}$$

$$s^t = f^t s^{t-1} + g^t \delta(b^i + U^i x^t + W^i h^{t-1}), \tag{5.4}$$

$$h^t = \tanh(s^t) o^t, \tag{5.5}$$

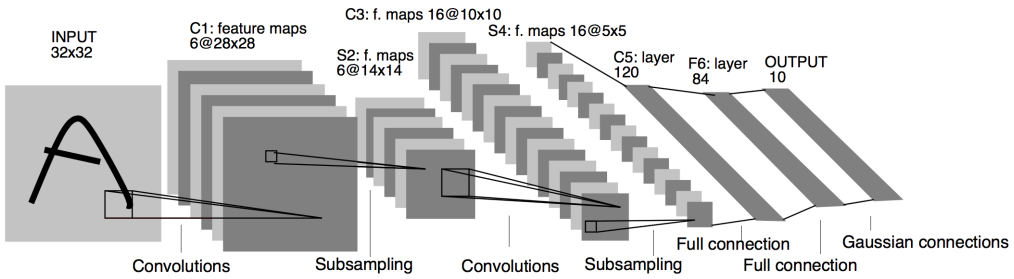

Figure 5.2: An example of a CNN: LeNet introduced by Yan LeCun [45]

where $\delta(x) = 1/(1+e^{-x})$ is the sigmoid activation function, $\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$ the hyperbolic tangent activation function, U^k, W^k and b^k with $k \in \{f, i, g, o\}$ are respectively the weight matrices and the bias vectors. The weight matrices and bias vectors are the parameters that are trained with the Back-Propagation Through Time (BPTT) that is a gradient-based technique for training this type of neural networks.

Convolutional Neural Networks (CNNs) [44] (see Figure 5.2) are deep neural networks for processing grid-like topology data (i.e., image data). Indeed, also the skeleton data can be mapped into an image, but the proper representation has to be investigated. A CNN can be thought of as a hierarchy of one, two or more convolutional modules that progressively learn higher-level features followed by one or two full connected layers that classify the extracted features. The main characteristic of a CNN are the sparse connectivity, the parameter sharing, and the equivariant representations. In detail, with respect to a traditional Artificial Neural Network (ANN) that has each input node connected to each output node, the CNN, instead, typically has sparse weights making the kernel smaller than the input. Moreover, while the traditional neural network has a multiplication between each element of the input and each element of the weights matrix, in the CNN, each member of the kernel is used at every position of the 2D input grid; it means that we learn only one set of parameters instead of different sets for every location. Finally, the CNN has equivariance to translation, but not to scaling or rotation. Typically, a convolutional module is composed by:

- a convolution layer that is a bank of affine transformations of input or in

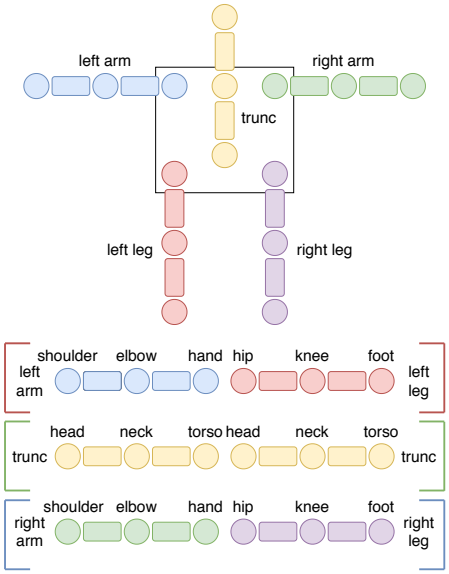

Figure 5.3: An abstract diagram of the proposed three-dimensional matrix for human pose representation

other words a bank of convolutional filters (also called kernels) applied on the 2D grid input;

- a detector layer that applies a non-linear activation function (typically the rectified linear unit - RELU);
- a pooling layer that reduces the input size (therefore it reduces the number of parameters) and improves the statistical efficiency.

The weights of the kernels in the convolutional layer are the parameters to learn to perform the training with stochastic gradient descent. Thanks to sparse connectivity and parameter sharing, the number of CNN's weights is reduced compared to the feed forward network.

Our initial aim was to automate also the extraction of the spatial features considering all possible connections between the skeleton joints. However, I found a reduced and concise representation that could well describe the human pose.

To be efficiently applied for action recognition, the first step is the transformation of the input data, the coordinates (x_i, y_i, z_i) of each of the ith joint of the human body at time t extracted by an RGB-D camera. Here, a novel representa-

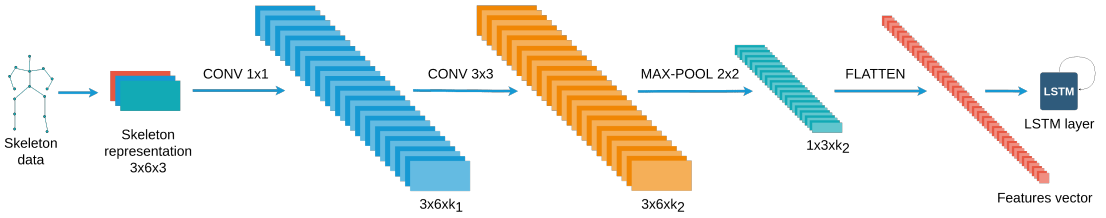

Figure 5.4: Combination of a CNN for automatic features extraction from the skeleton representation and an LSTM

tion of the joints values is proposed. Given the vector $f = [x_1, y_1, z_1, ..., x_N, y_N, z_N]$ of N skeleton joints, I combine these features in a three-dimensional matrix considering the spatial dependencies between the limbs. I have built a three-dimensional matrix to be invariant to translation, rotation, and scale. This matrix is the representation of the posture and it is the input of the CNN that can automatically extract the spatial features. The input is composed of three matrices referring to data related to the left arm (a_l)/leg (l_l), the trunk (t), and the right arm (a_r)/leg (l_r) of the human skeleton joints for each frame. Every considered limb is constituted by three joints each. For example, in the case of the left arm, the three joints are the left shoulder ($a_l[0]$), the left elbow ($a_l[1]$), and the left hand ($a_l[2]$). In the case of the left leg, the three joints are the left hip ($l_l[0]$), the left knee ($l_l[1]$), and the left foot ($l_l[2]$). The same is for the right arm and leg. In the case of the trunk, we have the head ($t[0]$), the neck ($t[1]$), and the torso ($t[2]$). The aim is to recognize the spatial dependencies between the limbs. Therefore, I model the following matrices representation:

$$\begin{bmatrix} a_l[x_0] & a_l[x_1] & a_l[x_2] & l_l[x_0] & l_l[x_1] & l_l[x_2] \\ a_l[y_0] & a_l[y_1] & a_l[y_2] & l_l[y_0] & l_l[y_1] & l_l[y_2] \\ a_l[z_0] & a_l[z_1] & a_l[z_2] & l_l[z_0] & l_l[z_1] & l_l[z_2] \end{bmatrix}$$

$$\begin{bmatrix} t[x_0] & t[x_1] & t[x_2] & t[x_0] & t[x_1] & t[x_2] \\ t[y_0] & t[y_1] & t[y_2] & t[y_0] & t[y_1] & t[y_2] \\ t[z_0] & t[z_1] & t[z_2] & t[z_0] & t[z_1] & t[z_2] \end{bmatrix}$$

$$\begin{bmatrix} a_r[x_0] & a_r[x_1] & a_r[x_2] & l_r[x_0] & l_r[x_1] & l_r[x_2] \\ a_r[y_0] & a_r[y_1] & a_r[y_2] & l_r[y_0] & l_r[y_1] & l_r[y_2] \\ a_r[z_0] & a_r[z_1] & a_r[z_2] & l_r[z_0] & l_r[z_1] & l_r[z_2] \end{bmatrix}$$

Figure 5.3 shows an abstract diagram to explain the disposition of the limbs in our proposed three dimensional matrix for human pose representation and feature extraction with the CNN. Each limb is composed of three joints and represents the rows of the three coordinates x, y, z. From this matrices representation, CNN learns the spatial features that involve the spatial limb correlations.

The proposed CNN is a three layers deep network (see Figure 5.4). The three matrices representation of the posture is given as input to the first convolutional layer. It sizes $3 \times 6 \times 3$ and has a set of kernels of size 1×1 and stride 1 to consider the spatial limb dependencies. Since the kernels size 1×1, the first convolutional layer linear recombines the weights based on the input feature maps as a parametric pooling layer. Therefore, its output sizes $3 \times 6 \times k_1$ and it is the input of the second convolutional layer. The second layer has a set of kernels of size 3×3 with stride 1. Its output sizes $3 \times 6 \times k_2$ where k_2 is the number of kernels. A max-pooling layer of size 2×2 with stride 2 halves the resolution of the third layer output and its output sizes $1 \times 3 \times k_2$. The size of 2×2 instead of the size of 3×2 is due to consider more information of the coordinates x and y than the information of coordinate z. A final layer flattens the output of the third layer concatenating the values in a vector with a length of $1 \cdot 3 \cdot k_2$.

The features extracted by the CNN are the input of the LSTM to identify the temporal dependencies of the change of the postures during the instance sequence. Hence, the LSTM layer takes as input a sequence of CNN output accumulating the temporal dependencies between each frame of the video. The LSTM input is a feature vector that contains the concatenation of the weight matrix. The LSTM is composed of a single layer and a number of neurons equal to the number of the feature vector extracted from the CNN. A full-connected layer with a softmax activation function classifies the activities performed in the video from the extracted features of the LSTM. The softmax function outputs a vector that represents the

probability distribution of a list of classes. It is usually used as the final layer of classifiers based on artificial neural networks. Given a feature vector x, the softmax function is as follows:

$$\delta(x)_i = e^{x_i} / \sum_{j=1}^{K} e^{x_j} = 1$$

5.2 Testing with dataset CAD-60

In this paragraph, I first introduce the dataset used for the experimental evaluation. Then, I describe the configuration of the proposed models and the results. Specifically, I compared the CNN-LSTM model based on our 3D skeleton representation with an architecture composed only by an LSTM layer to highlight the possible contribution of using CNN and the proposed joint matrix representation in accounting for spatial dependencies. Moreover, I will discuss our results in comparison to other state-of-the-art approaches tested on the same dataset.

Table 5.1: Number of 140 frame instances for each environment and each activity class

Environment	Class	User 1	User 2	User 3	User 4	Total
bathroom	**brushing teeth**	1212	1536	1644	1441	**5833**
	random + still	2684	2074	2729	2785	**10272**
	rinsing mouth with water	1607	1307	1364	1726	**6004**
	wearing contact lenses	557	1137	544	822	**3060**
bedroom	**drinking water**	1448	639	1171	1390	**4648**
	opening pill container	332	546	204	595	**1677**
	random + still	2684	2074	2729	2785	**10272**
	talking on the phone	1386	691	1149	1169	**4395**
kitchen	**cooking (chopping)**	1426	1525	1615	1771	**6337**
	cooking (stirring)	1207	1210	1328	1696	**5441**
	drinking water	1448	639	1171	1390	**4648**
	opening pill container	332	546	204	595	**1677**
	random + still	2684	2074	2729	2785	**10272**
living room	**drinking water**	1448	639	1171	1390	**4648**
	random + still	2684	2074	2729	2785	**10272**
	relaxing on couch	1308	1358	1240	1714	**5620**
	talking on couch	1542	1400	1573	1673	**6188**
	talking on the phone	1386	691	1149	1169	**4395**
office	**drinking water**	1448	639	1171	1390	**4648**
	random + still	2684	2074	2729	2785	**10272**
	talking on the phone	1386	691	1149	1169	**4395**
	working on computer	1126	1391	1083	1523	**5123**
	writing on whiteboard	1653	1498	1458	1653	**6262**

5.2.1 Dataset

Our project aims at recognizing the ADL to monitor the daily activities of elderly people. In this direction, I use the Cornell Activity Dataset (CAD-60) for training and testing the deep networks. The CAD-60 [81] is composed of 60 RGB-D videos captured by a Microsoft Kinect, with twelve activities performed in five environments. These videos are accomplished by four subjects, two males and two females, with one left-handed. The 12 labeled activities are: rinsing mouth, brushing teeth, wearing contact lenses, talking on the phone, drinking water, opening pill container, cooking (chopping), cooking (stirring), talking on couch, relaxing on couch, writing on whiteboard, working on computer. The CAD-60 dataset has two more activities (random and still) which are used together for classification assessment on testing sets. The 5 environments are office, kitchen, bedroom, bathroom, and living room. The dataset is made up of RGB and depth images, and the tracked skeleton. 15 skeleton information is extracted for each frame. The total number of videos is 68: 17 videos for each user.

I decided to use a temporal sliding window for considering all the contiguous frames, unlike [82] where they used a deep learning approach by selecting one frame every six frames of the videos to reduce redundancy and complexity. The smallest video of the CAD-60 is of 147 frames, therefore, I have set the instances of 140 frames (e.g., I obtain 8 instances with a video of 147 frames). Thus, the input sequence to the CNN-LSTM and LSTM models sizes 140 frames. Further considerations on the choice of the 140 frames window size can be found in the results of the CNN-LSTM model. In all model configurations, the validation set is 33% of the training set.

Table 5.1 shows the frequency distribution of the instances extracted from the CAD-60 dataset with 140 frames for each instance. In Table 5.1, I considered the environment and the activity class performed by each user. Note that the numbers of the instance are not balanced between the 13 activities. In particular, the "random + still" activity has a number proportional to the sum of the other activities for classification assessment.

5.2.2 Data Pre-processing

The number of skeleton joints tracked in CAD-60 is 15. 11 joints have both joint orientation and joint position while 4 joints have only the joint position. I considered only the joint positions of the 15 joints. To train our model on 140 frame instances, a temporal sliding window was applied. For each 140 frame instance, I have performed three pre-process steps for the coordinates of the skeleton joints as follows:

1. **Symmetrization.** Since in the dataset, there is one left-handed person, for each subject, I also considered mirrored skeleton data. To mirror the skeleton sequences, I took the opposite values of the x coordinate that are on the horizontal axis. In other words, the point coordinates $J = (x, y, z)$ become $J_{new} = (-x, y, z)$. This step doubles the number of dataset instances.

2. **Translation.** I set the midpoint between the points of the torso, left and right shoulder, left and right hand as the origin of the coordinates system. Once the midpoint was calculated, it was subtracted from the coordinates of the joints to have the midpoint as the center of the skeleton pose. For example, if I have a joint $J = (x, y, z)$ and the midpoint is $J_{mid} = (x_{mid}, y_{mid}, z_{mid})$, the new joint will be

$$J_{new} = (x - x_{mid}, y - y_{mid}, z - z_{mid})$$

.

3. **Normalization.** I compute the mean and the standard deviation for each instance to normalize the translated data on a new origin using the standard score: $J_{new} = (J - \mu)/\sigma$. The new coordinates are calculated following the previous formula applied to each coordinate (x, y, z). For each coordinate c (x, y, z), the following equation applies on all the elements i of each sequence:

$$J_{new_{ci}} = \begin{cases} (J_{c_i} - \mu_c)/\sigma_c, & \text{if } \sigma_c \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

where μ_c is the mean of the whole 140 frame sequence on the c coordinate while σ_c is the standard deviation of the whole 140 frame sequence on the c coordinate.

5.2.3 Model Settings

The settings of the deep models have an important role in the gradient convergence, preventing over-fitting on this tiny dataset. I used the Glorot normal initializer [31], also called Xavier normal initializer for the initialization of the LSTM weights for each deep model. The experiments showed that the deep models performed well with a dropout set at 0.25 after the max-pooling layer of the CNN and at 0.5 on the LSTM layer. Dropout [3] is a regularization technique to reduce overfitting in an artificial neural network by randomly pruning weights in an iterative process that leads to model improvement at each step of the process. CNN has 32 kernels in the two convolutional layers for a reduction of the number of parameters.

The CNN-LSTM model is compared with an LSTM model. The latter model is the same as the CNN-LSTM model without the CNN level. To make the comparison, I left the same LSTM layer configuration for both models. Both models receive an input sequence of human poses. Thus, in the LSTM model, we have consecutively a single LSTM layer, that extracts the temporal dependencies from the features vector $f = [x_1, y_1, z_1, ..., x_N, y_N, z_N]$ of N skeleton joints representing the human pose (without considering the spatial dependencies with a CNN), and a full-connected layer with a softmax activation function, that classifies the activities.

5.2.4 Implementation Details

I used the API of Keras library that is designed to simplify the development of the neural network. Originally developed on top of Tensorflow, now it is part of the Tensorflow library with the Tensorflow version 2.0. During the experiments, I ran the training and the testing on Keras version 1.2.2 with Tensorflow version

0.12.0.

5.2.5 Classification Results

Two different settings are considered in the original work on CAD-60 [79]: "New Person" and "Have Seen" settings. The most considered experimental setting in all the research works on CAD-60 is the "New Person" to guarantee the generalization of the classifier. The "New Person" setting is defined as a "Leave One Out (LOO)" cross-validation that is, the training set consists of three of the four people and the test set consists of the fourth one. In the "Have Seen" setting, the model is trained with half of the testing subject's data and the other half is included in the tests. In literature, the CAD-60 is split according to the considered environment. The final results are the average precision and recall among all the environments.

Tables 5.8 and 5.7 show the classification precision and recall of the proposed CNN-LSTM model in comparison with the LSTM model for each environment and for both the "New Person" and the "Have Seen" setting. First, we notice that the test set results of the CNN-LSTM model is better than the LSTM model and they are similar in both the settings, but with slightly better performance in the Have Seen setting.

Both models are expected to suffer from overfitting with a small training set. Especially in cases where there are a small number of training examples, the model may adapt to features that are specific only to the training set; therefore, in the presence of overfitting, the performance of the prediction on the training data will increase, while the performance in the test set will be worse. Hence, overfitting on data could have an impact more on the "Have Seen" setting, leading to better results, since training and testing are both obtained from the same subjects. Indeed, since performance in the "New Person" setting is very similar to the Have Seen case, we can consider overfitting as marginal.

From now on, we will make considerations only on the "New Person" setting.

LSTM Results

In the Living Room (97.4% and 96.8%) and the Kitchen (96.4% and 95.3%) environments, using the LSTM model, I achieved the best results in the "New Person" setting (see Tables 5.8 and 5.7) thanks to the recognition of the activity temporal patterns. Whereas, the worst results are achieved in the Office environment (91.63% and 89.49%). "Relaxing on couch" and "talking on couch" are discriminated at 100%, perhaps for the stationary character of the activities, while the LSTM model has difficulty in the disambiguation of "talking on phone" and "drinking water" in the Living Room and Office environment probably due to their similarities. For "writing on whiteboard", the LSTM model predicts "talking on phone" in the 20.6% of cases or "random + still" in the 3.6% of cases, thus its accuracy is lower than "talking on phone" accuracy.

Table 5.2: State-of-the-art results on CAD-60 dataset

	"New Person"	
Algorithm	Precision	Recall
Zhu W. et al. [86]	93.2%	84.6%
Faria D.R. et al. [29]	91.1%	91.9%
Shan J. et al. [78]	93.8%	94.5%
Parisi G.I. et al. [61]	91.9%	90.2%
Cipitelli E. et al. [12]	93.9%	93.5%
Khaire P. et al. [41]	93.1%	90.0%
Liu T. et al. [48]	97.97%	95.75%
Our LSTM	**95.07%**	**96.46%**
Our CNN-LSTM	**97.00%**	**98.00%**

CNN-LSTM Results

Considering the CNN-LSTM model, we have an improvement in the results compared to the LSTM model results. This is particularly evident in the Office environment. The lowest results are obtained in the Kitchen that, as previously dis-

cussed, has activities with periodic patterns as chopping. The CNN-LSTM model behaves better where the LSTM gets worse. We can see in Table 5.4 that the CNN-LSTM model has better results in precision in the Bathroom environment with the "random + still" (71.4% vs 94.0%), and in the Bedroom and the Kitchen environments with the "opening container" (82.3% vs 94.0% in the Bedroom and 80.7% vs 85.5% in the Kitchen). There are also different results in the Office environment in precision and recall respectively for the "talking on the phone" (78.2% and 96.3% vs 80.5% and 95.5%) and "writing on whiteboard" (89.4% and 75.7% vs 94.3% and 85.3%).

The overall activity confusion matrix, presented in Figure 5.5, shows the results in the "New Person" setting with the CNN-LSTM model. We can see that "cooking (stirring)", "drinking water", "random + still", "rising mouth with water", "writing on whiteboard" have lower accuracy than the other activities considering only the 140 frames as an instance.

Thanks to the representation of the skeleton with a 3D matrix, the results obtained with the CNN-LSTM model improves in comparison with LSTM. To evaluate the impact of the proposed approach, different combinations of input matrix have been texted leading to lower performance. For example, by inverting the left leg with the right arm, so to have in the first matrix the two arms, and in the last one the two legs, we got 92.74% of precision and 92.30% of recall against 95.40% and 94.38% of the proposed 3D matrix representation.

5.2.6 Statistical Hypothesis Test

In general, the model that best predicts unseen data might be the model with the maximum accuracy or minimum error for classification or regression problems. We can trust the model selected with the maximum accuracy or minimum error by applying a statistical hypothesis test. I applied the McNemar's test to check if the slightest differences we have between the CNN-LSTM model (97.00% of precision and 98.00% of recall) and the LSTM model (95.07% of precision and 96.46% of recall) are significant. The function takes the contingency table as an

Table 5.3: Results of our approach using different frame window on CAD-60 dataset with "New Person" setting

	"New Person"	
Model	Precision	Recall
LSTM on 50 frames	**91.21%**	**89.13%**
LSTM on 100 frames	**93.08%**	**91.55%**
LSTM on 140 frames	**95.10%**	**93.88%**
CNN-LSTM on 50 frames	**90.02%**	**88.89%**
CNN-LSTM on 100 frames	**92.22%**	**90.54%**
CNN-LSTM on 140 frames	**95.40%**	**94.38%**

argument and returns the calculated test statistic and p-value. The McNemar's test strongly confirmed that the CNN-LSTM model was significantly better than the LSTM model ($\chi^2 = 136026, p - value < 0.0001$) at a 95% confidence interval. In short, the results of the CNN-LSTM models were statistically significant at a significance level of 0.05.

5.2.7 Window Size Results

Let us now consider the possible impact on performance of the instances' window size. In order to do so, I made additional experimentation considering other frame windows: 50 and 100 frames. The results are shown in Table 5.3. With respect to 140 frames, as expected, considering fewer frames yields a decrease in performance (precision and recall). However, in view of the application of the proposed approach in real settings, fewer frames can still be considered since achieving good performance.

5.2.8 Comparison with the SoA

The CNN-LSTM model achieves, in the average, 95.4% and 94.4% on precision and recall. In Table 5.2, I reported our average results with respect to other approaches in the literature. I must emphasize the fact that I get such results

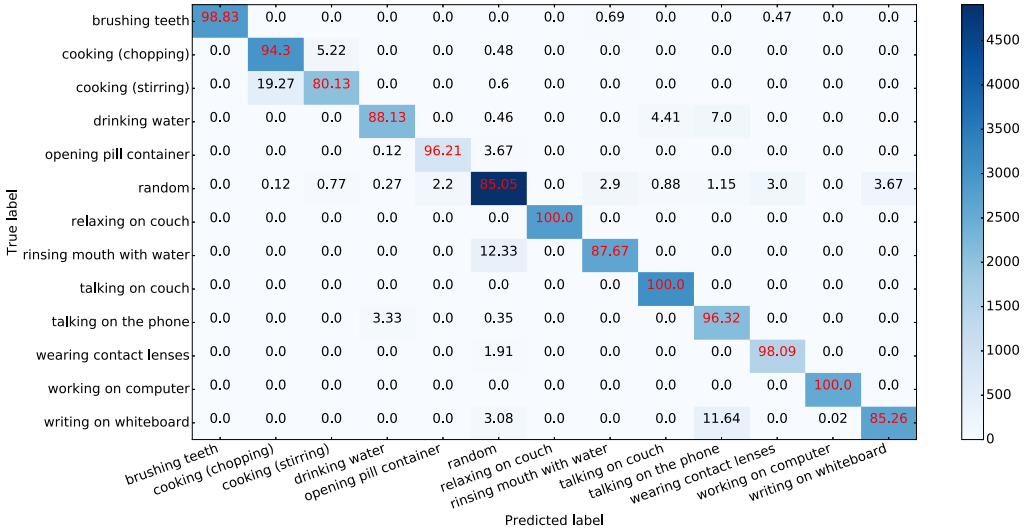

Figure 5.5: Overall activity confusion matrix in "New Person" setting with the CNN-LSTM model on 140 frames window

considering instances of 140 frames, while all the other works, reported in Table 5.2, considered the activity recognition on the entire videos. The shortest video is of 147 frames while the longest video is of 1961 frames. The average number of frames is about 1181 frames with 595 for standard deviation. Hence, our approach achieves a better performance with respect to all the other cases only considering small video sequences and skeleton data only. The only exception is the work of [48].

Applying the proposed model on the entire videos with the "New Person" setting, I obtained 96.46% of recall and 95.07% of precision with the LSTM model and I obtained 98.00% of recall and 97.00% of precision with the CNN-LSTM model reaching such state-of-the-art results in activity recognition on the CAD-60 dataset. Such results are obtained with a sliding window of 140 frames applied to each video, and by considering, for each classification result, only the output with an accuracy greater than 80%. The result of a classification process is then the most recognized activity. For example, on a video of the activity "drinking water" formed by 1448 instances of 140 frames I considered only the results of classifications with a probability greater than 80%. I obtained 1291 instances that are classified as "drinking water", 12 as "random + still" and 41 as "talking on

the phone". The predicted activity is therefore "drinking water" as it has been predicted more times over the entire video.

The comparison is made on the state of the art applied to the CAD-60 dataset. The classification in these SoA works is performed on the entire frame sequence of each video using manual features extraction and classic machine learning algorithm. The latter essentially involves the extraction of the characteristic poses of an activity using mainly clustering to select the significant poses that best describe the activity performed. I want to emphasize instead that the results I have obtained on the single instances are not comparable with the other works. On the contrary, the results obtained by applying the sliding window on the entire video are comparable. Moreover, as a difference of the SoA, we have carried out an automatic extraction of the features that is the basis of the potential of deep learning models. However, a pre-processing phase, which does not include feature selection, is necessary to train and run neural network models.

On average, only 4% of the frames for each video were discarded due to lower accuracy. Only two videos, regarding the third user, were not correctly recognized. Respectively, in the Kitchen environment, the "cooking (stirring)" activity was classified as "cooking (chopping)" and, in the Office environment, the "writing on whiteboard" activity was classified as "talking on the phone". I must emphasize that the third user is left-handed and the "cooking (stirring)" and the "cooking (chopping)" as the "writing on whiteboard" and the "talking on the phone" are very similar if we consider the movement of the human skeleton.

Considering the confusion matrix reported in Figure 5.5, we can observe that, although on average some activities have a lower recognition rate, I reached 98% of recall and 97% of precision on the entire videos with 140 frames sliding window approach. In this case, I supposed that some instances, i.e., sub-sequences of the videos, are the most likely to provide relevant information to correctly identify an activity while others are not. Indeed, this issue has to be taken into account when performing online recognition on sequences with a small number of frames.

5.2.9 Real Settings Configuration

The UPA4SAR project aimed at assisting and monitoring elderly people in their homes. Hence, we conducted the experimentation in real houses of the participants. 7 patients participated in the trials interacting with the robot for 2 weeks each. The experiments were performed by the robot in full autonomy, without the presence of an operator. For privacy and security reasons, it was not possible to save any video or audio and the robot had no internet access during the experimentation.

For training the network, we collected data from real patients during preliminary experiments in a laboratory resembling a house environment. The considered activities were, "talking and relaxing on the couch", "watching tv", "working on PC", "ironing", "making coffee", and "talking on the phone".

The robotic system used for experimentation consisted of a Sanbot robot and an Intel NUC (Intel NUC 8i7BEH2, Intel Core i7-8559U 4,5 GHz, 16 GB RAM, 250 GB SSD) for the execution of artificial intelligence algorithms that required computing power. During the daily experiments, a Workflow Manager, running on the Intel NUC, planned and scheduled the activities to be performed by the robot.

Among the activities, at particular times during the day, the robot was requested to monitor the user activity in order to check whether a specific activity was being performed by the user or not. This request was followed by the user search. The robot searching for the user positioned itself in front of the user and, once identified the user through facial recognition, the robot recorded 10 seconds of video, sending the frames to the Intel NUC to extract the skeleton poses. From the extracted skeleton poses, I applied a sliding window of 140 frames and I classified the activity performed on each instance. The recognized activity is the one with the highest number of recognitions from the ones with the confidence greater than 80%. In case the recognized activity was not the one "expected" the robot performed the recognition process three times leading eventually to a dialogue with the user in the case of mismatch.

The running time for a single 140 frames classification was about 0.015 seconds on the Intel Core i7-8559U 4.5GHz, while 2.42 seconds for processing the whole 10 seconds of data. Classification data cannot be reported because for privacy reasons videos were not saved and so it was not possible to get a ground truth.

5.3 Testing with our sampled dataset

In this section, I introduce the dataset recorded during some laboratory experiments described in [68] [27]. Next, I describe the proposed classifier model and the results obtained. The analysis of the experiments is concerned with how to approach the users and monitor their behavior. The intent of these works is to monitor the user and prevent the robot from distracting the user from the task at hand. During the experiments, I recorded videos from both environmental cameras and the robot's camera.

5.3.1 Dataset

Our goal is to recognize ADLs to monitor the daily activities of elderly people. Thus, we want to use only the robot's camera to reduce the intrusiveness of environmental cameras and the costs involved. In the first analysis, I created models of daily activity classifiers by training them on a public dataset such as CAD-60. The CAD-60 dataset was recorded using the Microsoft Kinect, a depth camera, allowing 15 joints of the 3D skeleton to be extracted. The robot used during experimentation is Pepper from SoftBank Robotics. The 2D camera placed on Pepper's top head recorded 10-second videos with a resolution of 320x240px and a framerate of 30 fps. Even with a low resolution it is possible to extract the joints of the human skeleton. Since the recorded videos from the front camera do not have the depth information, I estimated the skeleton joints with OpenPose [8]. The number of skeleton extracted is 36. The number of sequences related to the activity videos are 242. For each sequence I associated the following attributes: file name, activity, distance of observation, angle of the robot approach, number

Figure 5.6: Some sampled activities: talking on the phone (up), watching TV (center), ironing and making coffee (bottom).

of gazes, average duration, number of pose changes. The activities recorded in our dataset are working on PC, talking on the phone, watching TV, making a coffee, ironing, talking on couch with another person correspondingly labeled as PC, phone, TV, coffee, iron, couch (Figure 5.6). Trials were performed on a sample of 21 Italian senior volunteers (males = 12, females = 9), with an age between 53 and 82 years (M = 61.0, SD = 7.6) and years of education ranging from 8 to 18 (M = 12.5, SD = 3.6). Each participant had no prior experience interacting with robots.

5.3.2 Data Pre-processing

The number of skeleton joints tracked in our dataset is 36. Each joint is represented by (x,y) coordinates defined in pixels, which indicate the position of the point within the video frame image. The sliding time window applied on the videos is 140 frames. I choose this width according to previous experiments performed on

the CAD-60 dataset in [28]. For each instance consisting of 140 video frames, I applied a normalization on the coordinates according to the following equations:

$$x_{new} = x/width - 0.5$$

$$y_{new} = y/height - 0.5$$

where (x,y) are the coordinates of the joints while width and height are the width and height of the image resolution respectively.

5.3.3 Model Settings

The proposed classification model consists of a one-layer deep network. This layer consists of an LSTM, a deep neural network capable of automatically extracting features from a temporal sequence. The number of units set for the LSTM layer is 72. The classification layer is instead constituted by a full-connected network with 6 nodes corresponding to the 6 classes. The activation function is softmax. The calculated error for gradient descent is the categorical crossentropy. I used the optimization algorithm RMSprop in order to converge more quickly the training of the network. RMSProp (Root Mean Square Propagation) is a method in which each parameter (or weight) is updated based on the learning rate. First the current mean is calculated in terms of the root mean square,

$$v(w,t) = \gamma v(w,t-1) + (1-\gamma)(\nabla Q_i(w))^2$$

where, γ is the forgetting factor. And the parameters are updated as,

$$w = w - \frac{\eta}{\sqrt{v(w,t)}} \nabla Q_i(w)$$

5.3.4 Implementation Details

For the development of the classifier based on a deep neural network I used the API of the Keras library which is now part of the Tensorflow library. The version of Tensorflow used is version 2.2.0.

5.3.5 Classification Results

Unlike the previous experimentation described in section 5.2, the activities in our dataset were not evaluated based on the environment where that activity is performed, but are classified together using a single model. The setting applied for the classification of these 6 activities is the "New Person" as in the original work on CAD-60 [79]. This setting is none other than the "Leave One Out (LOO)", i.e., each set, consisting of the activities performed by a single person, is used once as a test set while the remaining sets constitute the training set. Figure 5.7 shows the confusion matrix based on the results obtained with 21 training epochs. The labels "PC", "phone", "TV", "coffee", "iron", "couch" correspond to the following activities, respectively: working on PC, talking on the phone, watching TV, making a coffee, ironing, talking on couch. We can see that "PC" and "couch" are recognized with an accuracy that exceeds 90%. The activity "TV" is exchanged in 18% of the cases with "couch". Conversely, the "couch" activity is identified as "TV" in 10% of the cases. This result is very feasible since they are very similar and since talking on the couch and watching TV are both activities performed on the couch. A similar reasoning can be done for the activities "coffee" and "phone". In fact "coffee" is classified in the 24% as "phone". Other relevant misclassifications we have for the activity "iron" which is interpreted by the classifier as "phone" in 12% of the cases and as couch in 10% of the cases. Although the challenge of classifying these activities performed daily at home is very difficult, often due to the difficulty of disambiguating very similar activities, we can be satisfied with the result. Overall, the average accuracy obtained is 82% which results as a good accuracy.

5.4 Additional Application about Robot Gesture Imitation performed by autistic children

The deep network-based method discussed so far for activity recognition is easily applied to gesture recognition as well. This section describes my work abroad carried out in collaboration with Sheffield Hallam University. We envision a gesture

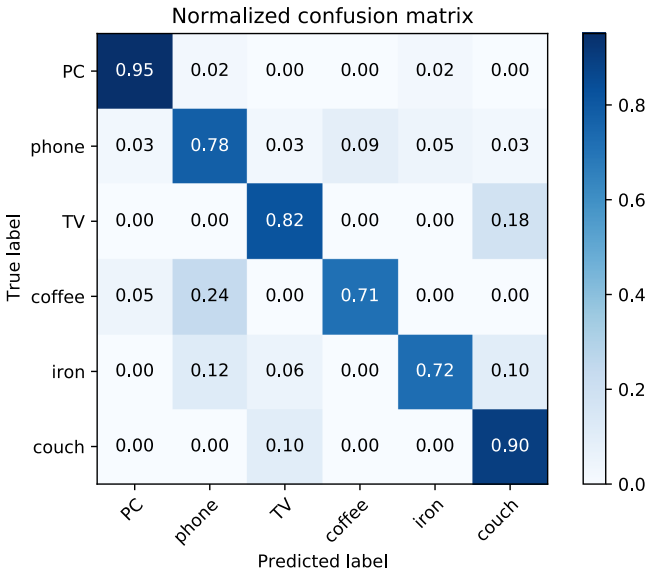

Figure 5.7: Normalized confusion matrix of the classification results on our dataset. The label activities are PC, phone, TV, coffee, iron, couch corresponding respectively to working on PC, talking on the phone, watching TV, making a coffee, ironing, talking on couch with another person.

recognition method based on deep neural networks. We wanted to identify the success or failure of the robot gesture imitation performed by the children. The aim of the proposed approach is to automatic extract the human skeleton pose of a video sequence and to automatic extract the temporal features between the different poses. Then, the resulted features are classified by a full-connected layer.

In literature there are many works that involves the gesture recognition with OpenPose, manual feature selection and classical machine learning algorithm. In [75], the authors extracted the human pose using OpenPose and recognizing the gestures with Dynamic Time Warping (DTM) and One-Nearest-Neighbor (1NN) from the time-series. Other works use instead more devices to better identify gestures with deep networks. In [50], they obtained 3D skeletal joint coordinates from 2D skeleton extraction with OpenPose and the depth from a Microsoft Kinect 2. Then, the 3D coordinates are used to detect the gesture using a CNN classifier. This system was employed for real-time human-robot interaction. Our intent is to reduce the number of devices yielding the built-in camera of NAO robot and

to recognize the child gestures from a sequence of 2D poses with a deep neural networks.

5.4.1 Proposed Method

The proposed method is divided into three steps. In the first step, each frame is computed by OpenPose [8] that is a real-time pose estimator. OpenPose returns the human pose in a reasonable time which depends on the computational power, extracting the pose from the image by a deep network based on the CNNs. See for instance Figure 5.8. Nevertheless, OpenPose was able to extract the human joints even if they are lacking. After gathering data from each video, transformed in human pose sequences with 18 joints, I normalized data according to the following equations that are applied for each joint (X, Y) assuming that the image centre is the origin $(0, 0)$:

$$X = \lfloor X + 0.5 * width \rceil; Y = \lfloor Y + 0.5 * height \rceil$$

where $width$ and $height$ are the image dimensions of the video.

In the second step, the human poses extracted from each video frame is given as input to a deep model based on LSTMs like in [28]. This model automatic extracts the temporal features of the poses sequence. I used 84 and 66 units respectively for the first and the second LSTM layer for "Already Seen" setting while 80 and 64 units for "Leave Child Out" and "Interleave" settings. The number of epochs was 300 to train the different models. The kernel initializer was the Xavier uniform initializer and the optimization algorithm for gradient descent was Adam.

The final step consists in classifying the gestures by a full-connected layer. During the experiments, 207 videos of about 1.10 minute and about 10 fps were recorded for six children. The gestures are four: "kiss", "clap the hands", "greeting", "raise the arms". I also added a "failure" class to label the imitation failures.

Then I trained our model with different configurations using a sliding window approach with one and two steps. We can also find a step approach in [56] and in [28] where the authors combine the results with different steps, considering

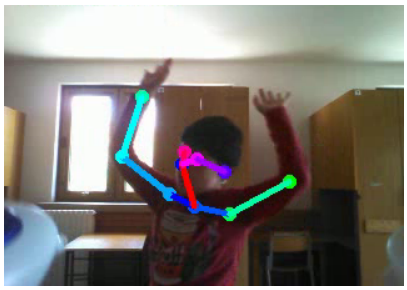

Figure 5.8: The frame video shows a child with his skeleton joints recognized by OpenPose [8].

different temporal scales, in contrast to us who do not combine the different steps. I used a sliding window of 5, 10, 15, 20, 25 sequence frames. The input of the model is composed by a sequence of human skeleton joints normalized according to the image dimensions and the label of the gesture performed by the robot ("kiss", "clap the hands", "greeting", "raise the arms"). The output is one of the four gesture labels or the label "failure" in case the child fails to imitate the robot.

The deep model based on LSTMs is composed by two LSTM layers that take in input the pose sequence. The features extracted from the sequence is concatenates with the gesture label encoded using the one-hot-encoding process that which refers to the gesture performed by the robot.

5.4.2 Settings

Three evaluation settings are proposed to assess the results of our approach: **Already Seen**, proposed [80] as "have seen", in which the training data is composed by five children and a half of the sixth child's data that is taken randomly; the test data is the remaining of the sixth child's data; **Leave Child Out**: the model was trained on five children and tested on the sixth; in literature we can find the same configuration named as "new person" or "leave-one-out crossvalidation" [80]; **Interleave**, similar to the "Leave Child Out" setting, but the gestures of different children were interleaved to take into account the significantly different quality and efficacy of the gesture executions.

5.4.3 Comparison with classical ML methods

I compared the type of approach proposed with classical machine learning methods using Weka. I have tested these algorithms both with and without normalization. The results show a general improvement in accuracy without normalization with respect to frame resolution. The pose sequences have been processed to extract the 5 most significant poses. I applied K-means, a clustering algorithm, to search for 5 clusters. Then I identified the 5 centroids that represent the 5 most significant poses that identify the sequence of the gesture. The 5 poses extracted for each instance are the samples of our ML classifier training dataset. I used the following classification algorithms which are models of supervised learning to compare our proposed approach:

- Bayesian Network is a probabilistic model that represents a set of stochastic variables with their conditional dependencies using a DAG (direct acyclic graph);
- HMM (Hidden Markov Model) is a Markov chain in which states are not directly observable and is widely used in the recognition of the time pattern of time series;
- Naive Bayes is a simplified Bayesian classifier that assumes assumptions of independence of characteristics;
- SVM (Support Vector Machine) is a model that represents data as points in space, mapping them in order to define the belonging of each data to a class;
- J48 is the implementation in Weka of the C4.5 algorithm, based on decision trees;
- Random Forest is a classifier obtained from the aggregation of multiple random decision trees;
- Random Tree is based on random decision trees.

5.4.4 Results

Table 5.4: Accuracy results for the three settings with a step of 1 frame using our method

Timestep	Setting	Accuracy (%)	Mean (%)
5	AlreadySeen	94.56	93.01
5	Interleave	92.49	
5	LeaveChildOut	91.99	
10	AlreadySeen	92.70	91.42
10	Interleave	91.42	
10	LeaveChildOut	90.14	
15	AlreadySeen	92.27	90.16
15	Interleave	88.85	
15	LeaveChildOut	89.35	
20	AlreadySeen	92.27	90.11
20	Interleave	88.85	
20	LeaveChildOut	89.21	
25	AlreadySeen	90.27	88.54
25	Interleave	87.13	
25	LeaveChildOut	88.21	

Three different settings, two different steps and five different timesteps are tested using our deep model obtaining the results showed in Table 5.4. I would like to emphasise the best results (see Tables 5.4 and 5.5) with a timestep of 5 and in general the tendency to overcome the 90.00% of accuracy. I want to underline the worst accuracy with "Interleave" and timestep 25 that is 87.13% of accuracy with step 1 and 87.06 of accuracy with step 2. The results gradually rise decreasing the timestep. Indeed, we have the best accuracy results in the setting "Already Seen" with 94.56% and 94.13% for step 1 and 2. I tested our system on a NVIDIA Jetson TX2 to explore the performance of our method in real-time gesture recognition, which can be used to personalise the intervention

Table 5.5: Accuracy results for the three settings with a step of 2 frames using our method

Timestep	Setting	Accuracy (%)	Mean (%)
5	AlreadySeen	94.13	92.49
5	Interleave	91.92	
5	LeaveChildOut	91.42	
10	AlreadySeen	94.56	92.13
10	Interleave	91.49	
10	LeaveChildOut	90.35	
15	AlreadySeen	93.42	91.16
15	Interleave	90.78	
15	LeaveChildOut	89.28	
20	AlreadySeen	91.85	90.11
20	Interleave	89.49	
20	LeaveChildOut	88.99	
25	AlreadySeen	92.99	89.44
25	Interleave	87.06	
25	LeaveChildOut	88.28	

with automatic adjustments to the children performance. The execution time on 1000 frames of OpenPose takes on average 0.13 ± 0.01 sec on each frame while our model takes on average 0.03 ± 0.00 sec on an entire sequence of 25 frames. I used the Mobilenet network in OpenPose algorithm to decrease the computational time on the Jetson TX2. I compared our method with classical machine learning algorithm. The classifier used are the following: SVM (Support Vector Machine), Bayesan Network, HMM (Hidden Markov Model), J48, Random Forest, Random Tree. The results of the SVM and the HMM algorithms are identical while in general all the other algorithms, except the Random Forest, have statistically worse results than the SVM and HMM algorithms at a significance level of 0.05. The Random Forest algorithm performs better than the SVM and the HMM only in the "AlreadySeen" setting. In short, our deep model have statistically better

Table 5.6: Accuracy results for the three settings with ML methods

Classifier	Setting	Accuracy (%)	Mean (%)
SVM	AlreadySeen	66.38	65,67
	Interleave	63.64	
	LeaveChildOut	66.99	
Bayesian Network	AlreadySeen	33.90	33,34
	Interleave	32.55	
	LeaveChildOut	33.56	
HMM	AlreadySeen	66.38	65,67
	Interleave	63.64	
	LeaveChildOut	66.99	
Naive Bayes	AlreadySeen	29.08	27,12
	Interleave	26.29	
	LeaveChildOut	25.98	
J48	AlreadySeen	62.84	58,53
	Interleave	56.89	
	LeaveChildOut	55.85	
Random Forest	AlreadySeen	72.03	66,24
	Interleave	62.15	
	LeaveChildOut	64.55	
Random Tree	AlreadySeen	61.41	53,96
	Interleave	48.80	
	LeaveChildOut	51.67	

results than all the tested machine learning algorithms at significance level of 0.05. One of the additional information we have is the behaviour of the robot that the child must imitate. In the final results I have noticed that they improve slightly by adding this information to the 5 poses extracted from the sequence of the gesture.

5.4.5 Discussion

A fundamental issue in the pose recognition was the motion of the NAO when performing gestures. Consequently, the video recorded by camera fixed on its forehead was unstable and made the recognition of the children gestures very challenging. Moreover the only device used in the experiment was the built-in camera of NAO

robot that has a low resolution (320 x 240) and a low frame-rate (10 fps) since the children are more comfortable with only the robot and without other cameras like Microsoft Kinect. In conclusion, I can say that the greatest difficulty was the lack of depth information since NAO is equipped with 2D camera and the occlusions due to movements of robot camera and child (depending of the level of ASD and ID). Small robotic platforms that are being used for robot-assisted therapy have usually limited sensors on-board, the actual resolution and frame-rate of NAO is are usually restricted to 320x240 and 10 fps due the limited computing capacity of the main processor and memory resources. Moreover, especially in the case of ID, children are unlikely to adhere to any imposed constraint like those typically required to maximize algorithm performance. Therefore, the challenge is to estimate the child's visual attention directly from the robot cameras, possibly without the need of external devices, such as high-resolution cameras and/or Microsoft Kinects (or equivalent), which can definitely increase the performance, but at the same time limit the portability of the system and make more difficult its actual integration within the standard therapeutic environment. The 2D poses of the children was extracted with OpenPose algorithm to deal the lack of the depth information. Another issue faced is that the dataset is unbalanced since it has multiple instances of children failures: the sum of gesture on the testset is about half of failures number. Although the results of the deep model with 1 step and 5 timestep are slightly better, in general the 2 step behaves well with the various timesteps. This is useful because in production stage the 2 step model can reduce the calculation time for the gesture prediction during the evaluation of children's imitation tasks, using an embedded AI computing device like the NVIDIA Jetson TX2 that has good performance and low power consumption. In particular, it will not be necessary to apply OpenPose to every frame, but every two frames, reducing by half the calculation time with acceptable prediction results. Finally, it is clear that the deep model has exceeded the results of the machine learning algorithms proposed for comparison.

Table 5.7: Precision (P) and Recall (R) of the LSTM and CNN-LSTM models on sequences of 140 frames for "Have Seen" setting on 140 frames window

		Have Seen			
		LSTM		CNN-LSTM	
Location	Activity	P (%)	R (%)	P (%)	R (%)
Bathroom	brushing teeth	96.6	100.0	100.0	99.8
	random + still	91.8	91.2	93.3	93.5
	rinsing mouth	92.8	84.4	95.8	88.0
	wearing lens	95.5	97.3	88.8	94.0
	Average	93.6	92.4	94.9	93.9
Bedroom	drinking water	96.7	89.9	95.8	93.0
	opening pill container	93.1	99.2	89.4	100.0
	random + still	99.5	90.8	99.8	95.2
	talking on phone	88.7	99.5	90.3	95.9
	Average	95.2	92.8	96.1	95.11
Kitchen	cooking (chopping)	80.0	100.0	87.5	100.0
	cooking (stirring)	96.8	66.4	99.7	77.1
	drinking water	92.5	100.0	95.3	99.4
	opening container	83.3	99.0	87.3	99.2
	random + still	100.0	90.1	98.9	94.5
	Average	92.9	89.9	95.3	93.7
Living room	drinking water	90.4	99.1	98.9	97.8
	random + still	99.7	91.9	100.0	97.7
	relaxing on couch	99.8	100.0	100.0	84.8
	talking on couch	100.0	99.5	100.0	100.0
	talking on phone	92.4	97.9	88.4	99.7
	Average	97.4	96.7	98.4	96.4
Office	drinking water	92.4	90.8	97.8	83.6
	random + still	100.0	84.3	99.1	91.2
	talking on phone	73.7	99.2	89.1	98.7
	working on computer	98.8	100.0	100.0	100.0
	writing on whiteboard	65.2	76.7	90.5	100.0
	Average	87.9	88.3	95.7	94.27
Overall Average		**93.3**	**92.0**	**96.1**	**94.7**

Table 5.8: Precision (P) and Recall (R) of the LSTM and CNN-LSTM models on sequences of 140 frames for "New Person" setting on 140 frames window

		New Person			
		LSTM		CNN-LSTM	
Location	Activity	P (%)	R (%)	P (%)	R (%)
Bathroom	brushing teeth	96.8	100.0	100.0	98.7
	random + still	71.4	90.5	94.0	93.4
	rinsing mouth	93.9	92.3	94.6	87.7
	wearing lens	89.7	98.9	89.9	98.0
	Average	94.9	94.1	94.9	93.9
Bedroom	drinking water	94.7	97.5	94.9	90.7
	opening pill container	82.3	97.3	94.0	96.9
	random + still	99.5	91.1	99.5	96.9
	talking on phone	91.2	95.6	89.3	94.4
	Average	95.2	93.9	95.9	94.7
Kitchen	cooking (chopping)	95.4	74.3	88.8	94.5
	cooking (stirring)	98.8	94.4	91.1	74.8
	drinking water	94.9	100.0	99.1	99.7
	opening container	80.7	92.4	85.5	95.5
	random + still	98.2	91.5	95.3	94.7
	Average	96.4	95.3	93.1	91.6
Living room	drinking water	91.7	95.2	99.8	93.1
	random + still	100.0	93.1	99.0	98.5
	relaxing on couch	100.0	100.0	100.0	100.0
	talking on couch	100.0	100.0	100.0	100.0
	talking on phone	91.0	98.1	92.5	99.2
	Average	97.4	96.8	98.7	98.5
Office	drinking water	91.7	93.5	95.0	90.5
	random + still	97.2	87.3	97.8	93.9
	talking on phone	78.2	96.3	80.5	95.5
	working on computer	100.0	100.0	100.0	100.0
	writing on whiteboard	89.4	75.7	94.3	85.3
	Average	91.6	89.5	94.3	93.0
Overall Average		**95.1**	**93.9**	**95.4**	**94.4**

Chapter 6

Field Experimentation

In this chapter, I show and discuss the final results of the experimentation conducted within the homes of elderly volunteers.

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. The experimentation was approved (no. 167/18) by the ethical committee of the University of Naples Federico II.

Multiple single-subject studies were planned to evaluate the usage of the personalized robot with respect to their acceptance, evaluation of the interaction, and system reliability. Indeed, individual differences between subjects in terms of cognitive impairments, but also education level, and psychological traits are big and relevant to the phenomenon of interest, so preventing to conduct statistical analysis that considers different individual factors in the evaluation. In any case, recruiting a large significant number of subjects can be difficult due to possible reluctance to be involved in such experimentation and to have an interaction with the robot that lasts for multiple days.

6.1 Collected Information

To personalize the daily assistive plans, several sessions with each participant took place to collect detailed information.

The neurologist team runs cognitive tests to provide their cognitive profile and classify the patient according to Subjective Memory Disorder, Mild Cognitive Impairment, and Alzheimer's Disease. They also provided their personal data including the education level in terms of the number of years of education. In addition, they collected information about their daily routine activities, their medication therapy plan, their entertainment preferences.

The psychologist team runs personality tests to provide the personality profile of each patient. For the personality profile the Neo Personality Inventory - 3 test (Neo-Pi-3) [51] measuring five personality traits Neuroticism, Extraversion, Openness to Experience, Agreeableness, and Conscientiousness was adopted. Only Neuroticism and Openness were considered for personalization in the project as the traits impacting the interaction with the robot and the technology acceptance. The scales of these traits were split into 3 ranges corresponding to a low, medium, high value of the considered personality trait.

6.2 Experimental Procedure

The testing of the robotic system was performed in the homes of 7 elderly volunteers. For each elderly person, data were collected on their cognitive abilities, personality tests were performed, and interviews were conducted. I report the main characteristics of each patient in the Table 6.1. The average is approximately 18 days experimentation, just over two weeks. The average age of volunteers is 72 years. The longest trial was performed with patient P6 while the shortest trial was performed with patients P2 and P3.

Experimentation of our robotic platform took place within the homes of the selected elderly. The daily care plan was performed every day for a number of days defined in Table 6.1. The number of days the robotic application was performed

Table 6.1: Overview of the participants. Acronyms: CDR (Clinical Dementia Rating), MCI (Mild Cognitive Impairment), SSC (Systemic Sclerosis).

ID	Gender	Age	Education	Memory Impairments
1	F	84	13	Alzheimer (CDR=2)
2	M	61	5	MCI
3	F	55	8	MCI
4	M	68	8	MCI
5	F	83	18	SSC
6	M	78	8	SSC
7	M	77	13	MCI

was related to patient availability and the plan agreed with the participants. For privacy reasons, no data such as video recordings or human skeletal tracking were collected. Instead, a variety of data were collected such as cognitive and personality tests, daily questionnaires, interviews, and data saved in log files pertaining to the execution of workflows and services. A personalized schedule is defined for each user that is tailored to the patient's needs, demands, and cognitive characteristics. The execution of the daily care plans is carried out by the robotic application without any remote control within the home environment. The daily routines that define the behavior of the robot are defined as "workflows". Each workflow is defined by a start time, a limit duration, and consists of an execution diagram that defines the scheduling logic for executing the services offered by the platform. The workflow scheduler is called "Workflow Manager". It communicates via the "Workflow Middleware" with the Server programmed in Nodejs using the "Socket.IO" library for communication. Each service is then executed by the platform according to the directives of the "Workflow Manager" which schedules the workflows to be launched during the day. The 5 workflows implemented for experimentation to monitor and stimulate the patient are as follows:

- WakeUp Monitoring: verifies the patient's awakening
- Lunch Monitoring: checks that the patient has had lunch

- Dinner Monitoring: verifies that the patient has had dinner
- Remind Medicines: verifies that the patient has taken his medication
- Cognitive Stimulation: cognitively stimulates the patient

The maximum number of care tasks delivered daily is 8. Some care workflows are performed multiple times during the course of the day according to the patient's needs, such as tasks to remember medications or suggest entertainment. Each workflow ends with either a notification or an alert. Each care task has a validity interval to ensure temporal adherence to the care plan. A workflow, to be effective, should not exceed the validity interval. For example, remembering medication plays a key role in care and should be done in the allotted time otherwise you must notify your caregivers or family members. On the other hand, for entertainment activities, the time interval is longer to allow for different content to be listened to or watched. Each workflow runs a set of services that are offered by the robotic application. The Table 6.2 shows the description of each service invoked by the "Workflow Manager".

6.3 Scheduling and Execution of the Personalized Daily Assistive Plan

Several log files were collected during the experimentation.These data show the results obtained from the experimentation and allow us to evaluate the daily performance of the robotic system. The 5 workflows defined for the daily care plan of each elderly patient are defined in the table 6.3 along with the total number of executions performed and the total number of days of execution.

In Table 6.3 some tasks have more executions than the number of days because they were executed multiple times per day while others have 0 executions. These tasks were not executed for several reasons:

- the task was not requested by the patient because it was unnecessary;

Table 6.2: Services Description

Service	**Description**
Find User	It verifies the patient's presence in the home
Check Health State	It checks the patient's health status
Suggest Cognitive Activity	It suggests to the patient an activity between game, video and music
Suggest Physical Activity	It suggests physical activity to the patient
In Room Detection	It checks the presence of the patient in a room of the house
Pose Detection	It checks the patient's posture
Check Medicines	It checks that the patient has taken his/her medication
Remind Medicines	It reminds the patient to take his/her medication
Play Video	It delivers a video that reflects the patient's preferences
Play Game	It plays a game
Play Music	It plays music according to patient preferences
Alert	Notice to caregivers
Notify	Notification of success and failure of a given activity

Table 6.3: Number of executed workflows for each patients

Number of Days	14	11	11	17	25	25	15
Patient	P1	P2	P3	P4	P5	P6	P7
WakeUp Monitoring	17	10	7	11	8	0	0
Lunch Monitoring	9	11	16	18	14	0	9
Dinner Monitoring	6	0	12	6	11	0	9
Remind Medicines	4	16	18	18	23	103	2
Cognitive Stimulation	25	10	12	27	30	107	29
Total	61	37	58	69	78	210	49

- the task cannot be performed because of infrastructural problems due to the composition of the room, such as, for example, the presence of carpets too high for normal robot movement or the presence of delicate furniture;

- the task is not followed due to priority issues, such as in the case of the task of remembering to take medicine, which has higher priority than other tasks; this happens due to the fact that two tasks are not performed at the same time;

- the task of remembering the medication can be performed multiple times during the course of the day and therefore, due to the mandatory limitation of a minimum of 8 care tasks per day, other tasks have been put on the back burner.

In Table 6.4 it is reported the average duration of each type of assistive task for each patient mediated on the number of days of the experimentation. The results show that even though the corresponding workflows are composed of the same number of microservices, their execution time differ because of the different execution time of the FindUser microservice, as reported in Figure 6.1. This is mainly due to two reasons: the different environments where the robot is located required different times to locate the user, and the different number of relatives living in the house that could be detected by the robot delaying the time necessary to identify the patient.

Table 6.4: Average duration for each workflow type and for each patient

Workflow	P1	P2	P3	P4	P5	P6	P7
WakeUp Mon.	0:32:52	0:47:10	0:37:08	0:48:23	0:28:44	0:00:00	0:00:00
Lunch Mon.	0:40:22	1:05:21	0:49:15	0:34:13	0:25:14	0:00:00	0:42:36
Dinner Mon.	0:37:10	0:00:00	0:56:03	0:52:05	0:22:06	0:00:00	0:34:03
Remind Med.	0:31:23	0:59:28	0:41:59	0:56:50	0:27:16	0:42:13	0:48:35
Cognitive Stim.	1:05:03	1:38:48	1:05:47	1:17:37	0:52:55	0:58:47	1:01:22

The average duration of each workflow for each patient and the number of executed workflows confirmed that the adherence of the scheduled assistive tasks to the timing of the daily routines of patients. The adoption of the validity time for each planned task when they are scheduled for execution allowed to avoid

time overlapping, and hence failures, of the planned tasks, automatically managed without requiring remote control of the robotic application.

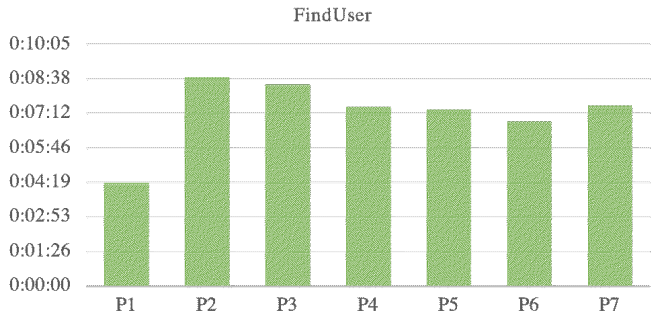

Figure 6.1: Average execution time for the *find user* service

The Cognitive Stimulation task reports the most variable execution times as reported in Figure 6.2. This is due to the different times required to play different types of entertainments as videos, music tracks, games, documentaries.

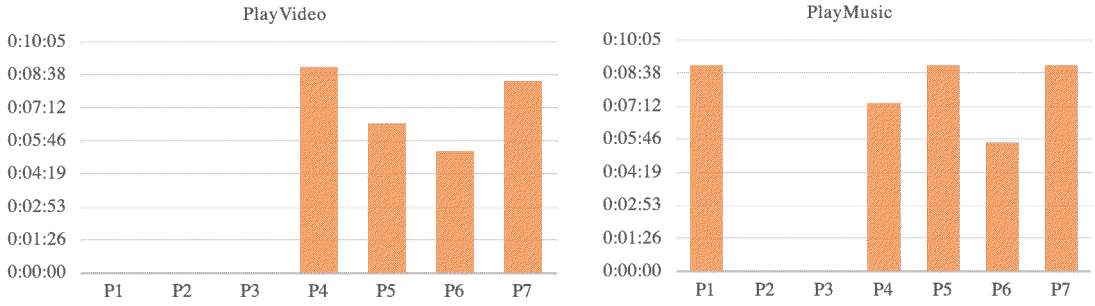

Figure 6.2: Average execution time for the *play video* and *play music* service

The data collected on the played entertainments are also analyzed as a feedback when users accepted and played them, to check whether the preferences they expressed during the interviews were valid during the experimentation, or if they were stimulated by the presence of the robot to experience different entertainments.

One of the most important assistive task to be executed with strict adherence to the daily routine is the RemindMedicine. The task consists in reminding the patient to take a medication in the case the patient forgot to do it, i.e. if CheckMedicine reports a "no" output. In such a case the assistive task is repeated

until either the CheckMedicine report a "yes" output, or the validity time expires requiring an alert to be sent to the caregiver for immediate action.

For this reason, as reported in Figure 6.3, the CheckMedicine and the RemindMedicine microservices are executed for a different number of times for each user depending on their responses when they were reminded to take the medicine. The P2 and P3 and P6 did not require a remind, while the patients P1, P4, P5 and P7 were to be reminded several times before actually taking medicines. These differences are due to the different cognitive status of the patients (the worse the memory impairment is the higher is the number of reminders), and the presence of caregivers or relatives that support them when crucial tasks have to be undertaken.

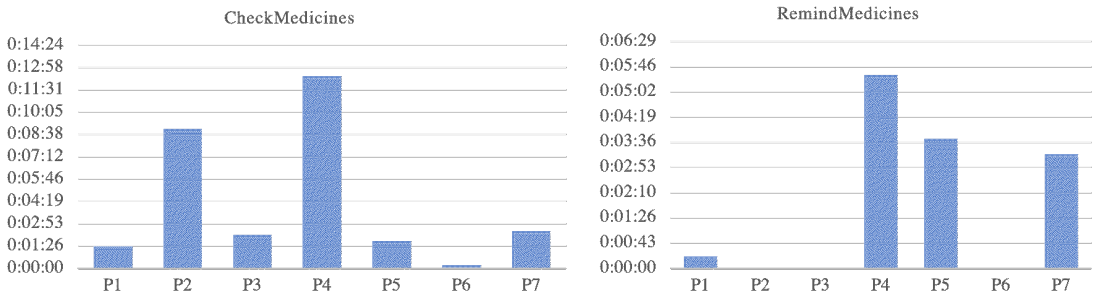

Figure 6.3: Average execution time for the *check medicine* and *remind medicine* task

Figure 6.4 shows the bar graph of the number of executions of the personalized and random workflows. Not including P6 who requested a personalized workflow plan, the other patients used both personalized workflows and random workflows. Considering the feedback reported in Section 6.5, only 2 patients, P4 and P7, reported negative feedback that did not involve the custom or random workflows, whereas the other 5 patients were very enthusiastic about their experience with the robot. If we look at the patients' daily feedback, no difference can be found between personalized and random workflows. The available data do not allow to reach significant conclusions about the differences between the personalized and random workflows and need to be supported by further studies on a larger number of patients.

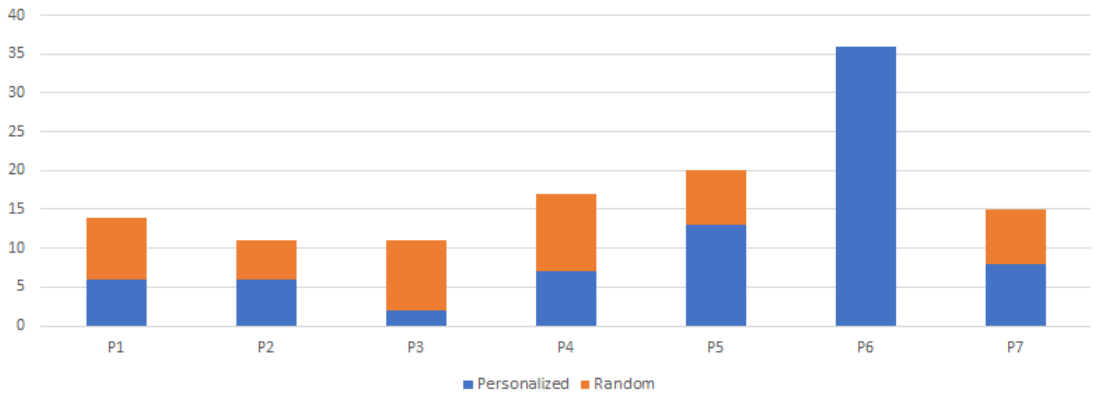

Figure 6.4: Number of personalized and random workflows for each patient

6.4 Results on Acceptance

During the experimentation, subjective information on how the interaction with the robot was perceived during the day was gathered through a very simple questionnaire, filled every day by the patient, that was asked to select one of the three emoticons representing their satisfaction degree with the following metrics: Sad=1, Neutral=2, Happy=3. Results showed that out of 7 patients, 4 evaluated the interaction with an average value above 2 (neutral) and 2 below (see Table 6.5).

Table 6.5: Average value of the daily subjective evaluation for each patient

Patient ID	P1	P2	P3	P4	P5	P6	P7
Average evaluation	2.8	2.3	2.1	1.5	2.8	2.7	1

Moreover, to evaluate the acceptance of the developed system with the user, the UTAUT questionnaire was used [36]. The questionnaire has been translated in Italian. The translation was examined at a consensus meeting, back-translated, and approved at a second consensus meeting. A comprehension test was carried out in a subgroup of 15 individuals.

UTAUT questionnaire aims at evaluating the user intentions to use a new technology and it consists of 41 items and explores 12 constructs: Anxiety (ANX), Attitude (ATT), Facilitating Conditions (FC), Intention to Use (ITU), Perceived

Adaptability (PAD), Perceived Enjoyment (PENJ), Perceived Ease of Use (PEOU), Perceived Sociability (PS), Perceived Usefulness (PU), Social Influence (SI), Social Presence (SP) and Trust (TR). The Likert scale to score the items ranges from 1 to 5. The SPSS software version 26 was used to analyze the data and calculate the statistics.

Table 6.6: Cronbach's Alpha values after removing items

Construct	α	**Construct**	α
ANX	1.0	PEOU	0.745
ATT	0.879	PS	0.816
FC	-	PU	0.889
ITU	0.977	SI	0.816
PAD	0.612	SP	0.876
PENJ	0.884	TRUST	0.998

First of all, we calculated the Cronbach Alpha (CA) [15] coefficient to estimate the internal consistency of each construct. Cronbach's alpha, α (or coefficient alpha), measures reliability, or internal consistency. Cronbach's alpha identifies how closely related the elements of a test are as a group. The formula for Cronbach's alpha is:

$$\alpha = \frac{N \cdot \bar{c}}{\bar{v} + (N-1) \cdot \bar{c}}$$

where N is the number of items, $\bar{c}$ is the average covariance between item-pairs amd $\bar{v}$ is the average variance. For ATT, ITU, PENJ, PU, SI, TRUST we had an α considering all the items above 0.8 that indicated an high level of internal consistency. For ANX, PAD, PEOU, PS, SP some items has been removed to reach a sufficient reliability. FC construct was removed as not reliable.

The descriptive statistics (mean, minimum, maximum, standard deviation) for the 12 constructs are reported in Table 6.7. For each construct, the result has been divided with respect to the number of items for the construct. We consider a positive perception of a participant is assumed when the construct score is greater than 2.5, while a negative perception is when average score is lower than 2.5.

Table 6.7: UTAUT results after the interaction with the robot

Construct		Avg	Std	Min	Max
ANX	Anxiety	4.43	1.51	1.00	5.00
ATT	Attitude	4.19	0.96	2.33	5.00
FC	Facilitating conditions	3.43	0.79	2.50	4.50
ITU	Intention to use	2.14	1.14	1.00	3.67
PAD	Perceived adaptability	3.00	0.96	2.50	4.50
PENJ	Perceived enjoyment	4.09	1.09	1.80	5.00
PEOU	Perceived ease of use	2.64	1.14	1.00	4.00
PS	Perceived sociability	3.00	1.20	1.00	4.33
PU	Perceived usefulness	3.67	1.58	1.00	5.00
SI	Social influence	4.14	1.11	2.00	5.00
SP	Social presence	2.39	1.39	1.00	5.00
TR	Trust	3.64	1.49	1.00	5.00

Table 6.8: Significant Pearson correlations between UTAUT contructs

	ANX	ATT	ITU	PENJ	PS	PU	SI	TR
ANX								.781*
ATT				.851*	.787*		.990**	
ITU						.869*		
PENJ		.851*			.890**	.938**	.869*	
PS		.787*		.890**		.910**	.814*	
PU			.869*	.938**	.910**			
SI		.990**		.869*	.814*			
TR	.781*							

The Shapiro-Wilk test on construct values showed that the data is normally distributed. Hence, to better understand relationships between constructs, we calculated the Pearson's correlations for parametric values. Once significant correlations were found between variables, the regression analysis was used to confirm the predictive role of one factor (predictor variable) to another (dependent vari-

able). Two-tailed correlation results are reported in Table 6.8.

6.5 Patients Interviews

A crucial source of information was gathered through interactions between the researchers responsible for the experimentation and the patients both via informing telephone calls, and friendly exchange of opinions at the end of the experimental period. It was difficult to formalize the obtained information, but since these feedback were very important to understand patients' perceptions during the experimentation we report here the main collected results.

Only 2 patients, P4 and P7, reported negative feedbacks about the experience with the robot mainly due to the lack of of vocal interaction, and to the expectation that the robot could be instructed to do tasks by using its monitor, or could move when they wanted. One of the patient declared to have skills in using the computer and hence being able to program it and to interact with it as with technological devices such as Alexa. In addition, the homes of these patients were quite small and not very suitable to allow the robot to easily move, and so they were disappointed by the few interactions due to the failures of some assistive tasks.

The other 5 patients were all very enthusiastic about the experience with the robot, and they all asked for a longer staying of the robot. They all declared at the end of the experimentation that the robot became like a friend making their time at home more pleasant, and making them feeling more occupied than usual. They were very thrilled by the fact that the robot recognized and interacted only with them, so making them feeling important for the robot. Above all, they appreciated the help they received in observing their medication therapy, and the possibility to have the entertainments they preferred (music tracks they liked that reminded them of their past times, recipes videos for who likes cooking, sport video for who likes sport, and so on).

In particular, the patient with the worse cognitive impairment (CDR=2) was very reluctant at the beginning almost refusing to have the robot at the first

sight. But when the robot started interactions everyday the attitude of the patient changed completely, and the robot was perceived as a family person.

The patients that had the robot for longer times (P5 and P6) were the most enthusiastic about it. This is an encouraging result since, even though supported by few data for deriving conclusions, it may suggest that the more used the patient becomes to the robot's presence, the higher its acceptance could be. In fact, one of them patient wrote an appreciation letter to the researchers thanking them for the special experience at the end of the experimentation. The other one developed a so close relation with the robot and expressed even worries when, during experimentation, the robot stopped for a day because it was out of charge (it was not recharged in time) and it did not move like it happened in the previous days. It was perceived like a real person not feeling well. The collected information about their feelings for the robot presence is in line with the UTAUT results that reported for these two patients an high value for the social presence construct, that is also higher compared to the ones of the other patients.

Chapter 7

Discussion

This chapter summarizes and shows some conclusions of the research experiments described in this thesis. Thanks to medicine and modern technologies in the field of health, life expectancy has increased considerably. For this reason, the elderly population has been increasing more and more and constitutes nowadays a considerable percentage of the world population. A key role in improving life expectancy can be played by technology. In particular, it can be an enabling tool for older people. Increasingly, older people lead a daily life alone, in some cases left to their own devices, with considerable difficulty in taking care of themselves. This happens in cases of people with dementia and is often the trigger for dementia due to the resulting social isolation, thus affecting their autonomy at home and making it difficult to perform the simplest daily activities. It would take a lot of intervention by family members or caregivers to solve this problem. One of the most serious cases of dementia is Alzheimer's disease, a disease that causes a progressive loss of memory and a worsening of cognitive functions. This disease affects both the elderly and their families who have to take care of them themselves or through a caregiver. In these cases, the elderly person even goes as far as not recognizing their own family members, or even, not recognizing places, the usefulness of objects and more. The progressive loss of short-term memory caused by Alzheimer's, puts the elderly person's life at serious risk and they need continuous daily monitoring. In recent years, much research has been done in the area

of socially assistive robotics (SAR). It can come to help and be a good solution to mitigate this problem, improving the lives of the elderly and his family members. Robots can be proactive agents that can, not only monitor the elderly for their safety, but also keep them company and stimulate them to slow down the progressive loss of cognitive functions. This thesis illustrates a low-cost robotic application for assisting the elderly. This work was supported by MIUR under the PRIN2015 research project "User-centered Profiling and Adaptation for Socially Assistive Robotics - UPA4SAR". The key features of the proposed robotic application are:

- acceptance of the robotic system through adaptation of the robot's behaviors based on the person's preferences and personality;
- low cost of the entire robotic platform for greater deployment in the homes of the elderly;
- modularity, robustness and scalability through the use of libraries that manage an event-driven communication to invoke microservices;
- ability to monitor activities through artificial intelligence algorithms;
- complete autonomy of the robotic system to execute daily care plans based on the person's needs;
- entertainment and companionship through multimedia content and games through interaction between the robot and the person;
- notification tool to remind when to perform certain activities during the day such as taking medicine

Comparing ourselves with other state-of-the-art projects on robotics of assistance for elderly people, we defined functional requirements following the characteristics described above. We then devoted ourselves to the analysis and implementation of a service-based robotic architecture. In particular, to make the architecture modular and scalable, I modeled its components as microservices.

Microservices allow the complex services offered by the system to be divided into many simple primitive functionalities [26]. Each microservice can constitute a single assistive action or be combined with other microservices to provide complex functionality. Adding new services is made easy by this type of microservice-based architecture. In addition, it is easier to customize services to the users' needs by invoking only those services that are needed. Nowadays, applications based on system virtualization such as the very popular Docker and the use of development frameworks such as Nodejs allow you to build scalable, secure and multi-tenant cloud services [42], greatly reducing costs, as demonstrated by companies such as Netflix, Reddit, Pinterest, and many more. In fact, many of the components developed required the use of Docker containers due to incompatibilities with the libraries of other components. Nodejs acted as a pivot to joust all the components divided into microservices. Such architecture can then be easily put into production via cloud services. However, given the fundamental requirement of patient privacy, everything was run during experimentation on a small server, the Intel NUC. Communication between the server and the robot was provided by a router, disconnected from the Internet, on a secure local area network. The communication between the user and the robot is done via an Android tablet that displays the robot's dialogues, either via text or voice, while the user can interact via the tablet's touch display. For privacy issues and to ensure security on the data processed by the robotic system, all services work off-line without any Internet connection. For this reason it has not been possible to exploit cloud-based services for voice interaction.

Our robotic system consists of a series of modules. It can be simplified in Figure 4.1 by considering that each of the following core components has associated complex services: "Robot", "Smartwatch", and "Server components" that communicate with the central node, represented by the "Nodejs Server (socket.io)". The central node is the main server script that runs in Nodejs and leverages communication through the Socket.IO library. Socket.IO uses an event-based protocol to manage communication between the server and clients. Thanks to the proposed

architecture, it is easy to add new functionalities or to transfer the entire robotic application to other technological environments. Among the many services implemented, I want to highlight the activity recognition service. This service is fundamental for the monitoring of the elderly person in order to guarantee his safety and verify the correct execution of a care plan. Therefore, I have created deep neural network models to classify ADLs (Activities of Daily Living). Classifying ADLs is a particularly difficult challenge to overcome. I tested models on the public CAD-60 dataset [81], chosen specifically for its sampled activities. I obtained very good results and compared them to the state of the art. Because the activities in the CAD-60 dataset do not fit all contexts and because it consists of activities performed by young people, we recorded during laboratory experiments the activities performed by older volunteers. The dataset obtained was used to train a model able to recognize the following 6 activities: working on PC, talking on the phone, watching TV, making a coffee, ironing, talking on couch with another person. The results obtained are good having reached an average accuracy of 82% with the "Leave One Out (LOO)" setting. If we look at the confusion matrix in Figure 5.7, we see that some tasks create misclassifications due to the fact that the movements are very similar.

Testing assistive robotics application in the real world and without the supervision of technicians remains a significant challenge despite the available technologies. The service-oriented adopted approach allowed to decouple the functioning of the developed microservices from both the design and implementation of the architecture of the complete robotic applications as well as from the collection of user data. In such a way it possible to perform in parallel testing of the system in the controlled environment of the University robotic laboratory, during the development phase of the project, while completing the implementation of all necessary functionalities. This was a fundamental step to improve the reliability of the robotic application to be then deployed in not controlled home environments.

In addition, the service-oriented approach proved to be a promising way in the direction of developing plug and play assistive applications crucial for personal-

izing home care in an application area where both the technological and service development progress at an incredible rate.

Most effort was devoted to guarantee the functioning of the system in a not controlled and supervised environment without relying on any remote control since network infrastructures may not be available in all the environments. Nevertheless, to have significant results from the interaction, it is necessary to obtain a significant maturity and dependability of the developed services in particular with respect to robot navigation and interaction capabilities. In particular, user localization in one of the more time consuming services in real home environments when not supported by different devices, not available for the requirements of low cost appliances and a low invasiveness and adaptability to different houses.

The technological constraints were not only due to cost and infrastructure limitations, but also to privacy and security reasons. In fact, no network connection from the house with outside was allowed, as well as no personal information from camera was allowed to be stored. These ethical issues pose challenges in the evaluation of results in the wild regarding the perception of the received assistance from the user, and the possibility to correctly evaluate the identification of the Activity of Daily Living.

Another challenge of the adoption of long term robot-application is the necessity of continuous user modeling due to both the rapid evolution of their cognitive impairment, and consequently also of their personality profile [20]. Even though the system design allows for the continuous adaptation of the robotics assistive behavior, of the delivered assistive tasks, and their timing, gathering correct information from patients may become difficult. As emerged by the opinions emerged from the patients' interviews, the time the robot is left at home may play a crucial role in its perception of usefulness by the users, and in the change that may occur in the perception over time. Of course, a study on this aspect was not undertaken since the times for the experimentation for each patient was constrained by the project timing and patient availability.

All collected data and their analysis suffer, as already pointed out, from the

difficulty of finding patients with the right conditions to be enrolled in such kind of experimentation, and from both the necessary time and technology availability.

Finally, the results on the patient's acceptability of the robot were influenced by the lack of the natural language understanding that was not included in the robot functionality since it required an outside internet connection.

Bibliography

[1] Koubaa Anis. Ros as a service: web services for robot operating system. 2015.

[2] Moez Baccouche, Franck Mamalet, and Christian et al Wolf. Sequential deep learning for human action recognition. In *Intern. Workshop on Human Behavior Understanding*, pages 29–39, 2011.

[3] Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26:2814–2822, 2013.

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[5] Carl Boettiger. An introduction to docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.

[6] Manuele Bonaccorsi, Laura Fiorini, Filippo Cavallo, Alessandro Saffiotti, and Paolo Dario. A cloud robotics solution to improve social assistive robots for active and healthy aging. *International Journal of Social Robotics*, 8(3):393–408, 2016.

[7] Radhia Bouziane, Labib Sadek Terrissa, Soheyb Ayad, and Jean-Franois Breth. A ros-based approach for robot as a service (web services based solution). *University of Biskra*, 2015.

[8] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[9] Yinong Chen, Zhihui Du, and Marcos García-Acosta. Robot as a service in cloud computing. In *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, pages 151–158. IEEE, 2010.

[10] Yinong Chen and Zhizheng Zhou. Robot as a service in computing curriculum. In *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, pages 156–161. IEEE, 2015.

[11] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR 2018*, 2018.

[12] Enea Cippitelli, Samuele Gasparrini, Ennio Gambi, and Susanna Spinsante. A human activity recognition system using skeleton data from rgbd sensors. *Computational intelligence and neuroscience*, 2016, 2016.

[13] Silvia Coradeschi, Amedeo Cesta, Gabriella Cortellessa, Luca Coraci, Javier Gonzalez, Lars Karlsson, Francesco Furfari, Amy Loutfi, Andrea Orlandini, Filippo Palumbo, et al. Giraffplus: Combining social interaction and long term monitoring for promoting independent living. In *2013 6th international conference on Human System Interactions (HSI)*, pages 578–585. IEEE, 2013.

[14] Serhan Coşar, Manuel Fernandez-Carmona, Roxana Agrigoroaie, Jordi Pages, François Ferland, Feng Zhao, Shigang Yue, Nicola Bellotto, and Adriana Tapus. Enrichme: Perception and interaction of an assistive robot for the elderly at home. *International Journal of Social Robotics*, pages 1–27, 2020.

[15] Lee J Cronbach. Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334, 1951.

[16] Claudia Di Napoli, Emanuela Del Grosso, Giovanni Ercolano, Federica Garramone, Elena Salvatore, Gabriella Santangelo, and Silvia Rossi. Assessing usability of a robotic-based aal system: A pilot study with dementia patients. In *WOA*, pages 59–64, 2019.

[17] Claudia Di Napoli and Silvia Rossi. A layered architecture for socially assistive robotics as a service. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 352–357. IEEE, 2019.

[18] Claudia Di Napoli, Luca Sabatucci, Massimo Cossentino, and Silvia Rossi. Generating and instantiating abstract workflows with qos user requirements. In *ICAART (1)*, pages 276–283, 2017.

[19] Claudia Di Napoli, Marco Valentino, Luca Sabatucci, and Massimo Cossentino. Adaptive workflows of home-care services. In *2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 3–8. IEEE, 2018.

[20] A. D'Iorio, F. Garramone, F. Piscopo, C. Baiano, S. Raimo, and G. Santangelo. Meta-analysis of personality traits in alzheimer's disease: A comparison with healthy subjects. *J Alzheimers Dis.*, 62:773–787, 2018.

[21] Jeffrey Donahue, Lisa Anne Hendricks, and Sergio et al Guadarrama. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[22] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. Microservices: yesterday, today, and tomorrow. In *Present and ulterior software engineering*, pages 195–216. Springer, 2017.

[23] Y. Du, Y. Fu, and L. Wang. Skeleton based action recognition with convolutional neural network. In *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 579–583, Nov 2015.

[24] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[25] Peter F Edemekong, Deb L Bomgaars, and Shoshana B Levy. Activities of daily living (adls). 2017.

[26] Giovanni Ercolano, Paolo D Lambiase, Enrico Leone, Luca Raggioli, Davide Trepiccione, and Silvia Rossil. Socially assistive robot's behaviors using microservices. In *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1–6. IEEE, 2019.

[27] Giovanni Ercolano, Luca Raggioli, Enrico Leone, Martina Ruocco, Emanuele Savino, and Silvia Rossi. Seeking and approaching users in domestic environments: Testing a reactive approach on two commercial robots. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 808–813. IEEE, 2018.

[28] Giovanni Ercolano, Daniel Riccio, and Silvia Rossi. Two deep approaches for adl recognition: A multi-scale lstm and a cnn-lstm with a 3d matrix skeleton representation. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 877–882. IEEE, 2017.

[29] Diego R Faria, Cristiano Premebida, and Urbano Nunes. A probabilistic approach for human everyday activities recognition using body motion from rgb-d images. In *The 23rd IEEE Intern. Symp. on Robot and Human Interactive Communication, RO-MAN*, pages 732–737. IEEE, 2014.

[30] David Fischinger, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann, Tobias Koertner, Astrid Weiss, Antonis Argyros, et al. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, 75:60–78, 2016.

[31] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[32] Google. Pose detection in the browser: Posenet model. (Date last accessed 10-May-2019).

[33] Horst-Michael Gross, Steffen Mueller, Christof Schroeter, Michael Volkhardt, Andrea Scheidig, Klaus Debes, Katja Richter, and Nicola Doering. Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5992–5999. IEEE, 2015.

[34] Horst-Michael Gross, Andrea Scheidig, Steffen Müller, Benjamin Schütz, Christa Fricke, and Sibylle Meyer. Living with a mobile companion robot in your own apartment-final implementation and results of a 20-weeks field study with 20 seniors. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2253–2259. IEEE, 2019.

[35] Tianyi Gu, Momotaz Begum, Naiqian Zhang, Dongpeng Xu, Sajay Arthanat, and Dain LaRoche. An adaptive software framework for dementia-care robots.

[36] Marcel Heerink, Ben Kröse, Vanessa Evers, and Bob Wielinga. Assessing acceptance of assistive social agent technology by older adults: the almere model. *International Journal of Social Robotics*, 2(4):361–375, 2010.

[37] Steffen Herbold, Alberto De Francesco, Jens Grabowski, Patrick Harms, Lom M Hillah, Fabrice Kordon, Ariele-Paolo Maesano, Libero Maesano, Claudia Di Napoli, Fabio De Rosa, et al. The midas cloud platform for testing soa applications. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*, pages 1–8. IEEE, 2015.

[38] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[39] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, Jan 2013.

[40] Michelle J Johnson, Megan A Johnson, Justine S Sefcik, Pamela Z Cacchione, Caio Mucchiani, Tessa Lau, and Mark Yim. Task and design requirements

for an affordable mobile service robot for elder care in an all-inclusive care for elders assisted-living setting. *International Journal of Social Robotics*, pages 1–20, 2017.

[41] Pushpajit Khaire, Praveen Kumar, and Javed Imran. Combining cnn streams of rgb-d and skeletal data for human activity recognition. *Pattern Recognition Letters*, 2018.

[42] M Kim, Ajay Mohindra, Vinod Muthusamy, Rohit Ranchal, Valentina Salapura, Aleksander Slominski, and Rania Khalaf. Building scalable, secure, multi-tenant cloud services on ibm bluemix. *IBM Journal of Research and Development*, 60(2-3):8–1, 2016.

[43] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[44] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[45] Yann LeCun, LD Jackel, Leon Bottou, A Brunot, Corinna Cortes, JS Denker, Harris Drucker, Isabelle Guyon, UA Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.

[46] Chuankun Li, Pichao Wang, Shuang Wang, Yonghong Hou, and Wanqing Li. Skeleton-based action recognition using lstm and cnn. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 585–590. IEEE, 2017.

[47] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-

regression recurrent neural networks. In *14th European Conference on Computer Vision – ECCV, Part VII*, pages 203–220. Springer, 2016.

[48] Tingting Liu, Jiaole Wang, Seth Hutchinson, and Max Q-H Meng. Skeleton-based human action recognition by pose specificity and weighted voting. *International Journal of Social Robotics*, 11(2):219–234, 2019.

[49] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. *arXiv preprint arXiv:1802.09232*, 2018.

[50] Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, Robin Passama, and Andrea Cherubini. Towards real-time physical human-robot interaction using skeleton information and hand gestures. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6. IEEE, 2018.

[51] Robert R. McCrae, Paul T. Costa Jr, and Thomas A. Martin. The neo-pi-3: A more readable revised neo personality inventory. *Journal of Personality Assessment*, 84(3):261–270, 2005.

[52] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. Affdex sdk: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*, pages 3723–3726. ACM, 2016.

[53] Philippe Merle, Christophe Gourdin, and Nathalie Mitton. Mobile cloud robotics as a service with occiware. In *2017 IEEE International Congress on Internet of Things (ICIOT)*, pages 50–57. IEEE, 2017.

[54] Maria Nani, Praminda Caleb-Solly, Sanja Dogramadzi, Tina Fear, and Herjan van den Heuvel. Mobiserv: an integrated intelligent home environment for the provision of health, nutrition and mobility services to the elderly. 2010.

[55] Dana Nau, Yue Cao, Amnon Lotem, and Hector Munoz-Avila. Shop: Simple hierarchical ordered planner. In *Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2*, pages 968–973, 1999.

[56] Natalia Neverova, Christian Wolf, Graham W Taylor, and Florian Nebout. Multi-scale deep learning for gesture detection and localization. In *European Conference on Computer Vision*, pages 474–490. Springer, 2014.

[57] Juan C Nunez, Raul Cabido, Juan J Pantrigo, Antonio S Montemayor, and Jose F Velez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.

[58] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[59] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018.

[60] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[61] German I Parisi, Cornelius Weber, and Stefan Wermter. Self-organizing neural integration of pose-motion features for human action recognition. *Frontiers in neurorobotics*, 9:3, 2015.

[62] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. Design and evaluation of a smart home voice interface for

the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing*, 17(1):127–144, 2013.

[63] David Portugal, Paulo Alvito, Eleni Christodoulou, George Samaras, and Jorge Dias. A study on the deployment of a service robot in an elderly care center. *International Journal of Social Robotics*, 11(2):317–341, 2019.

[64] David Portugal, Luís Santos, Paulo Alvito, Jorge Dias, George Samaras, and Eleni Christodoulou. Socialrobot: An interactive mobile robot for elderly home care. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 811–816. IEEE, 2015.

[65] Jürgen Pripfl, Tobias Körtner, Daliah Batko-Klein, Denise Hebesberger, Markus Weninger, Christoph Gisinger, Susanne Frennert, Hakan Eftring, Margarita Antona, Ilia Adami, et al. Results of a real world trial with a mobile social service robot for older adults. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 497–498. IEEE, 2016.

[66] Sekou L Remy and M Brian Blake. Distributed service-oriented robotics. *IEEE Internet Computing*, 15(2):70–74, 2011.

[67] Silvia Rossi, Luigi Bove, Sergio Di Martino, and Giovanni Ercolano. A two-step framework for novelty detection in activities of daily living. In *International Conference on Social Robotics*, pages 329–339. Springer, 2018.

[68] Silvia Rossi, Giovanni Ercolano, Luca Raggioli, Emanuele Savino, and Martina Ruocco. The disappearing robot: an analysis of disengagement and distraction during non-interactive tasks. In *2018 27th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 522–527. IEEE, 2018.

[69] Silvia Rossi, Giovanni Ercolano, Luca Raggioli, Emanuele Savino, and Martina Ruocco. The disappearing robot: An analysis of disengagement and distraction during non-interactive tasks. In *2018 27th IEEE International*

Symposium on Robot and Human Interactive Communication (RO-MAN), pages 522–527. IEEE, 2018.

[70] Silvia Rossi, Giovanni Ercolano, and Mariacarla Staffa. Towards an adaptive user monitoring based on personality and activity recognition. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 269–270, 2017.

[71] Silvia Rossi, Gabriella Santangelo, Martina Ruocco, Giovanni Ercolano, Luca Raggioli, and Emanuele Savino. Evaluating distraction and disengagement for non-interactive robot tasks: A pilot study. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 223–224, 2018.

[72] Silvia Rossi, Mariacarla Staffa, Luigi Bove, Roberto Capasso, and Giovanni Ercolano. User's personality and activity influence on hri comfortable distances. In *International Conference on Social Robotics*, pages 167–177. Springer, 2017.

[73] Pericle Salvini, Cecilia Laschi, and Paolo Dario. Design for acceptability: improving robots' coexistence in human society. *International journal of social robotics*, 2(4):451–460, 2010.

[74] Joe Saunders, Nathan Burke, Kheng Lee Koay, and Kerstin Dautenhahn. A user friendly robot architecture for re-ablement and co-learning in a sensorised home. *Assistive Technology: From Research to Practice (Proc. of AAATE)*, 33:49–58, 2013.

[75] Pascal Schneider, Raphael Memmesheimer, Ivanna Kramer, and Dietrich Paulus. Gesture recognition in rgb videos using human body keypoints and dynamic time warping. In *Robot World Cup*, pages 281–293. Springer, 2019.

[76] Ch Schroeter, Steffen Mueller, Michael Volkhardt, Erik Einhorn, Claire Huijnen, Herjan van den Heuvel, Andreas van Berlo, Andreas Bley, and H-M Gross. Realization and user evaluation of a companion robot for people

with mild cognitive impairments. In *2013 IEEE International Conference on robotics and automation*, pages 1153–1159. IEEE, 2013.

[77] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[78] Junjie Shan and Srinivas Akella. 3d human action segmentation and recognition using pose kinetic energy. In *IEEE International Workshop on Advanced Robotics and its Social Impacts*, pages 69–75. IEEE, 2014.

[79] Jaeyong Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *2012 IEEE International Conference on Robotics and Automation*, pages 842–849, May 2012.

[80] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *2012 IEEE international conference on robotics and automation*, pages 842–849. IEEE, 2012.

[81] Jaeyong Sung, Colin Ponce, and Bart et al Selman. CAD-60 and CAD-120. `http://pr.cs.cornell.edu/humanactivities/data.php`.

[82] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE Access*, 6:1155–1166, 2017.

[83] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[84] Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang. Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Transactions on Multimedia*, 2018.

[85] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3697–3703, 2016.

[86] Yu Zhu, Wenbin Chen, and Guodong Guo. Evaluating spatiotemporal interest point features for depth-based action recognition. *Image and Vision Computing*, 32(8):453–464, 2014.