SenseBot: A Wearable Sensor Enabled Robotic System to Support Health and Well-being

Luigi D'Arco[1,2], Huiru Zheng[2], Haiying Wang[2]

[1] Department of Computer Science DI, University of Salerno, Salerno, Italy
l.darco12@studenti.unisa.it
[2] School of Computing, Ulster University, Newtownabbey, Antrim, UK
{darco-l, h.zheng, hy.wang}@ulster.ac.uk

Abstract. The exponential growth of the technology industry over the past years has led to the development of a new paradigm. Pervasive Computing, which seeks to integrate computational capabilities into everyday objects so that they can communicate efficiently and perform useful tasks in a way that minimises the need for the end-users to interact with computers. Along with this paradigm, new technologies, such as wireless body area sensor network (WBASN) and robotics, have been rapidly growing. Such innovations can be used as health and wellness enablers. This paper introduces a system to support the health and well-being of people in both in-door and out-door environments. The system consists of an app, a robot and a remote server. The app and the robot can be connected to a wristband to monitor movements, physical activities, and heart rates. Furthermore, the app provides functions for users to record and monitor their calories intakes, and completed workouts. The robot is equipped with speech capabilities which, integrated with the emotion recognition algorithm, provide supportive feedback to the user. The proposed system represents an early step towards automated care in everyday life which opens the doors to many new scenarios, such as for elderly people who need help but live independently or also for people who would like to improve their lifestyles.

Keywords: Wearable Sensor · Robotic System · Emotion Recognition · Speech Recognition

1 Introduction

In 1991 Mark Weiser introduced his vision about the evolution of computer in the next century and the possibility of using computers and other devices in our lives without the user being aware of it [1]. At the time, this vision was too futurist, because the progress in computing was not so profound, however, the progress made in the last twenty years in both hardware and software has opened the doors to new concepts and approaches. The early big "computing machines" have been reduced to minimum levels enabling to embed computers in many parts of our environments. Now the term "pervasive computing" has replaced the old "ubiquitous computing" and that vision has become real.

Pervasive computing is typically associated with the race to miniaturise hardware with some degree of connectivity, intelligence and mobility [2][3]. With the growth of this paradigm a lot of applications which were initially available at a fixed location only have been transformed into ubiquitous applications, to be used wirelessly and flexibly at anytime and anywhere, and new technologies have been increased, such as the Wireless Body Area Sensor Networks (WBASNs) which defines an autonomous system consisting of several intelligent sensor nodes which monitor without hindering the daily life activities of users [4]. WBASNs have become one of the most promising technology for enabling health monitoring at home, especially for supporting older people's health and well-being, thanks to the low power consumption devices and the possibility to monitor activities, movements and vital human body signals of users continuously and remotely [5] [6]. The world's population is ageing, as provided by the World Health Organization [7] the proportion of the world's population over 60 years will nearly double from 12% to 22% between 2015 and 2050, and different governments are trying to promote home-based elder-care to reduce long-term hospital stays, home care services for elderly people. In several cases, however, to provide healthcare at home is not enough using a WBASN because patients can have physical problems or require additional help, so, new alternatives forms of healthcare are needed. One potential solution is the use of robotics which can provide tangible help for needy people [8] [9].
This paper proposes a solution to support the health and well-being of people in daily living, integrating data from wearable sensors with a robot. The system is composed of three heterogeneous sub-systems: a mobile application, a robotics system and a remote server. The mobile application allows keeping track of the user's everyday activities, such as food consumed and workouts performed, along with his extrapolated health details from a wristband. The mobile application allows also, to check the user trends by the user himself or by a designated parent. The robotic system is designed as an assistance agent capable of speaking, recognising voice commands and collecting fitness data from the user. The robotic system aims to recognise the user's emotion and consequently start a conversation. The remote server is the main source of storage for the system. It stores data from both the mobile application and the robotic system, subsequently, it performs data processing and reasoning. The collaboration between these three sub-systems enables the development of a system that is capable of collecting information from the user via multiple sources of information, offering more accuracy and reliability. The multiple sources of knowledge allow the system to operate in both indoor and outdoor environment, since they are not mandatory together, allowing a greater degree of independence for the user. The objective is to design a system that can be used by most people without any hindrance, keeping costs to a minimum and making the system as simple as possible.

This paper is structured as follow. Related works are described in Section 2. The architecture and implementation of the system are presented in Section 3. The tests are described and evaluated in Section 4, followed by the findings discussed in Section 5. The paper is concluded by a summary and future work in Section 6.

2 Related Work

Thanks to the various improvements in the area of robotics and the miniaturisation of wearable devices, different works can be found in the literature, that combine these two technologies to improve healthcare. Huang et al. [10] proposed an omnidirectional walking-aid robot to assist the elderly in the daily living movement. The robot is controlled during normal walking using a conventional admittance control scheme. When a fall inclination is detected the robot will immediately respond to prevent the user from falling. The fall detection is calculated using the assumption of the human Center of Pressure (COP) and through a wireless sensor, the Center of Gravity (COG) of the user can be approximated.
Goršič et al. [11] introduced a gait phase detection algorithm for providing feedback in walking with a robotic prosthesis. The algorithm is developed as a state machine with transformation rules based on thresholds. The algorithm is finally evaluated with three amputees, walking with the robotic prosthesis and wearable sensors. The studies in which wearable sensors and robotics meshed together are multiple and cover a large area but if we focus only on home healthcare, there are few studies in which it is treated.
Novel robotics and cloud-assisted healthcare system (ROCHAS) was developed by Chen et al. [12]. This study takes as its target users the empty-nesters. The system incorporates three technologies in terms of body area networks, robotics, and cloud computing. In particular, it consists of a robot with the speaking skills that allow the empty-nester to communicate with his/her children, several body sensors that can be deployed in or around the empty-nester, and a cloud-assistant healthcare system that stores and analyses the data provided by the robot. The system helps the empty-nester to be in touch with his/her children and at the same time allowing the children to be mindful of their elderly people's conditions.
Ma et al. [13] developed a healthcare system based on cloud computing and robotics, which consists of wireless body area networks, robots, software system and cloud platform. This system is expected to accurately measure a user's physiological information for analysis and feedback, which is assisted by the robot integrated with various sensors. To boost the viability of multimedia delivery in the healthcare system, this paper proposes a new scheme for transmitting video content in real-time via an enhanced User Datagram Protocol (UDP)-based protocol.

3 System Design and Implementation

This section will describe in detail the system architecture, dealing with the communication between the different subsystems.

The proposed system is composed of three main components: remote server (henceforth called PyServer), mobile application (henceforth called *MyFit*), and robot infrastructure (henceforth called PyBot). These three components work together to gather information from a person, store the information and take actions to help the person to be healthy. As can be seen in Fig. 1, the communication between these components is led by the HTTP protocol, and the MyFit and PyBot exploit a Bluetooth Low Energy (BLE) protocol to interface with the wearable sensor. The choice of the HTTP protocol is driven by the need for a protocol that can be suitable on multiple and heterogeneous devices in different locations. On top of the HTTP protocol are created different Representational State Transfer (REST) API architectures, as described below, to encourage the use of an understandable language for all the components, JSON (JavaScript Object Notation), which is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value) [14].

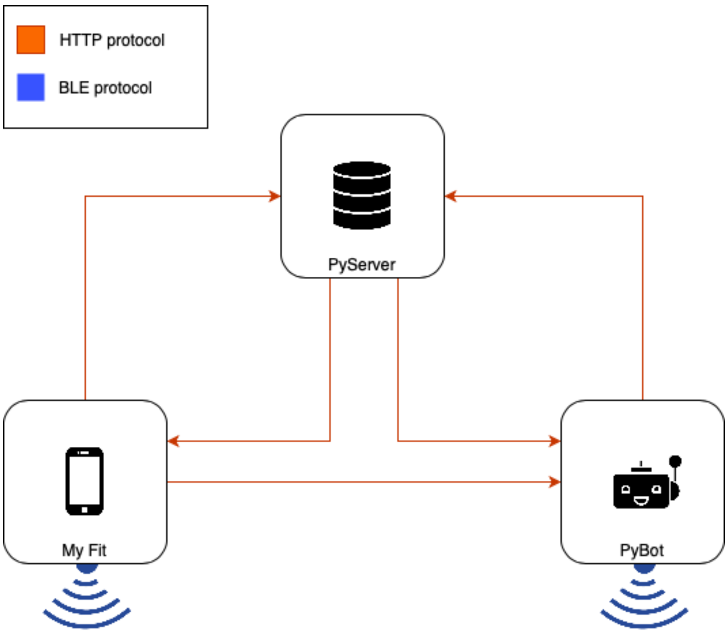


Fig. 1: System behaviour

3.1 MyFit

MyFit is a mobile application that can help the overall system to acquire new data from the user during the day. The app is divided into two versions: the *fit version* and the *robot version*.
The main purpose of the *fit version* is to gather information from wristband (steps done, distances travelled, calories burned, heart rates), instead, the main purpose of the *robot version* is to control remotely the PyBot. Leaving aside the

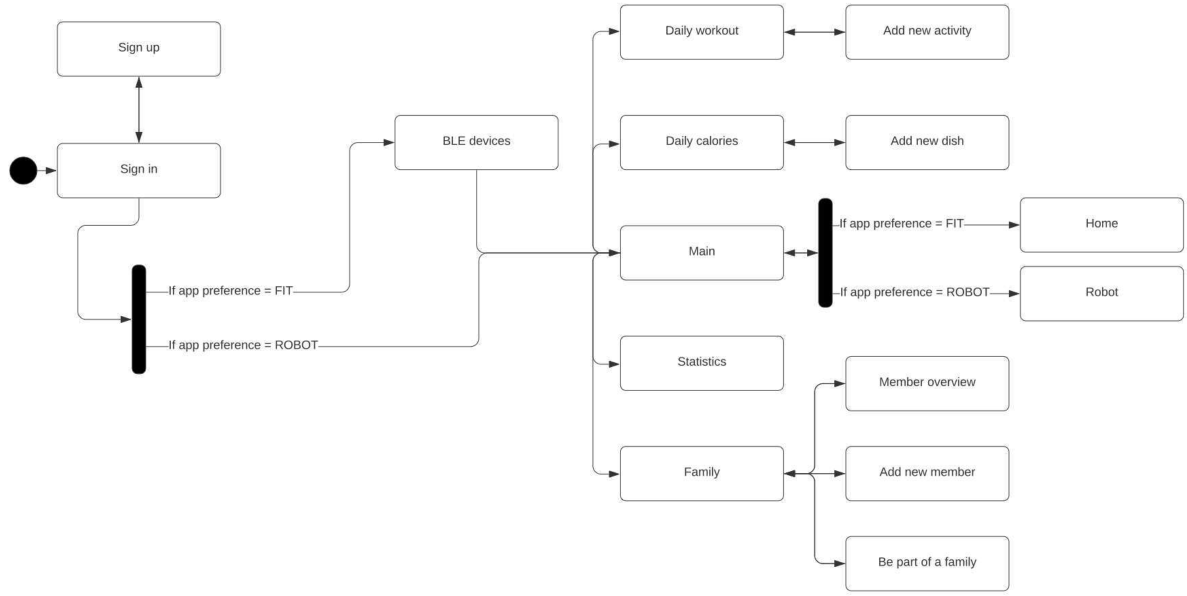
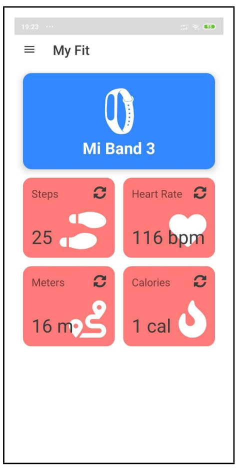


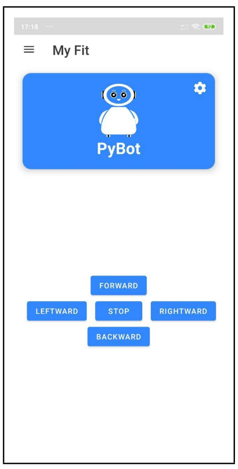
Fig. 2: MyFit: navigational path

differences in the main purpose, the other functionalities of the app are shared for both the versions. The entire navigational path of MyFit is shown in Fig. 2

The app can be accessed only after authentication, then the user, when starts the app, has to sign-in, if already register, otherwise he can sign-up and automatically he is sign-in. After the authentication step if the app run is the *fit version* the user has to choose the wristband from the list of available Bluetooth devices and then it is redirected to the main view where are shown all his fitness data, otherwise, if the app run is the *robot version* it is redirected to the main view where it can control the PyBot. Fig. 3 shows the different views that the user sees according to the version.



(a) fit version



(b) robot version

Fig. 3: MyFit: main views according to the versions

Once the user is in the main view, he can move among other views through the left navigation drawer, activated by the button on the top left.
The possible views are:

- *daily calories*: this view allows the user to see and to add the food eaten with the numbers of calorie, in the current day or the past days according to the meal.
- *daily workouts*: this view allows the user to see and to add the activities done with the amount of time spent, in the current day or the past days.
- *statistics*: this view allows the user to see the trend of its fitness data in the last 10 days.
- *family*: this view allows the user to see its family members[1], add a new one scanning his QR-code, be a family member of another user generating the QR-code and letting him scan it.

MyFit is developed for Android Devices, the devices supported are all the devices that run a version of Android between *Android 7.0* and *Android 10*. It is developed in JAVA and XML.

3.2 PyBot

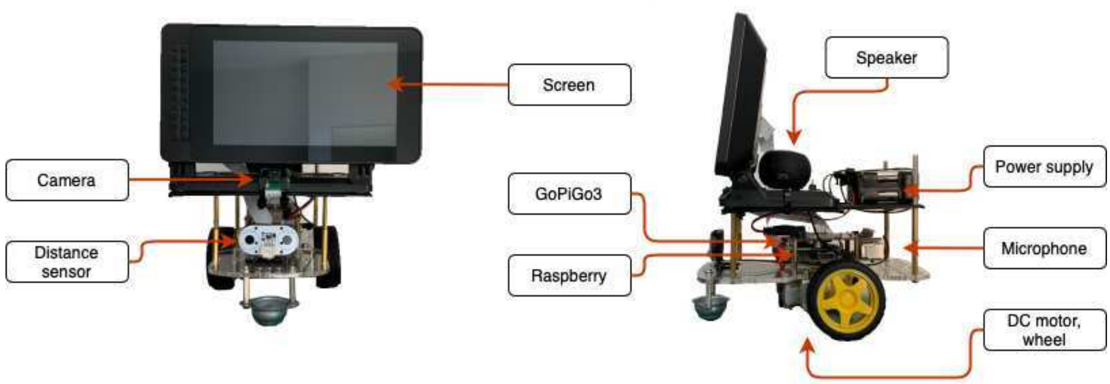


Fig. 4: PyBot: design

The physical design of the robot platform allows to interact in a friendly way with the user, to recognise the user's emotions, and to recognise the fitness activity made by the user. The robot is based on a design made by [17] with different updates. As shown in Fig. 4 the robot structure is composed of layers that can allow modularity so that in future improvements new components can be added easily. The main component of the robot, as well as the core of the robot, is a Raspberry Pi board, this choice is due to a need of maintaining the overall cost of the robot low to reach as many as possible people. Two DC motors

[1] family member: a user who you can check his fitness data trends

with two wheels are combined to allow the robot to move, allowing flexibility on which type of surfaces the robot can travel, thus restricting it to flat surfaces. Since the robot can move, it has incorporated a distance sensor to prevent it from colliding with objects on the front. The DC motors and the distance sensor are managed by the GoPiGo3 board. The robot has to communicate with the user in a friendly way, to reduce the gap between human and machine, for this reason, are integrated a screen that visualises information, a speaker, and a microphone. The robot has to, also, recognise facial emotion of the user, to allow this a Raspberry Pi Camera module is mounted on the front of it.

The robot platform can interact with the user and take decisions thanks to a python program. The program is built of four processes that run parallel on the machine: digital assistant, API manager, fit manager, emotion manager.

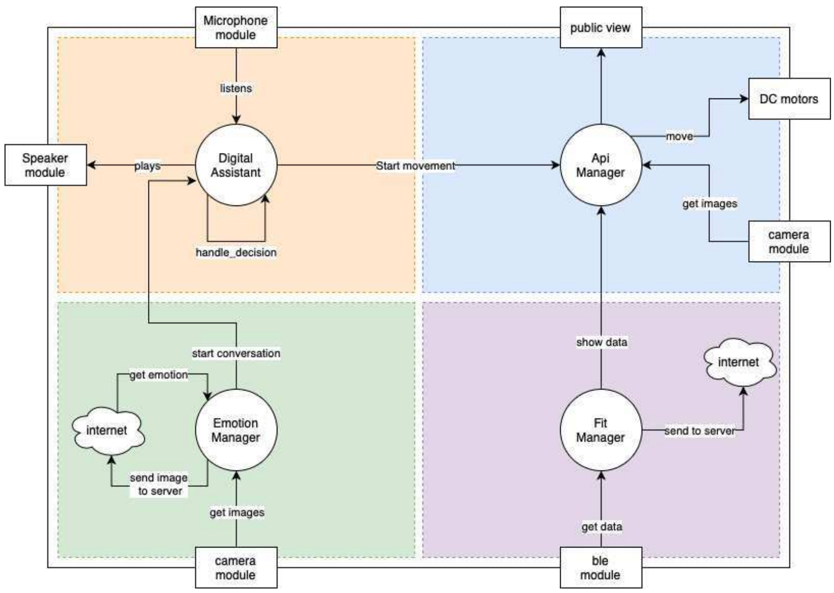


Fig. 5: PyBot processes

Digital Assistant This process is mainly responsible for the conversation with the user. The digital assistant can recognise and speak four different languages (English, Italian, Chinese, Spanish) but only one at a time, this can be set at the stat-up of PyBot.

When the PyBot starts up the digital assistant begins recording sound. Through the SpeechRecognition library [18] the recording is cleared by the ambient noises, and after thanks to its internal engine, GoogleSpeechRecognition, it tries to detect speech within the recording, if so the speech is converted to text, otherwise, the flow is stopped and the digital assistant starts to record again. When the text recognised is available a matching function between the preset voice commands and the text is applied. If the match has a positive result, the linked action is triggered, otherwise, the flow is interrupted and the digital assistant starts recording again. The triggered action is followed by a vocal response. The vocal response is performed by converting the English textual response connected to the action into the current language of the PyBot using the googletrans library [19], then the translated response is converted to voice thanks to the gTTS (Google Text-To-Speech) library [20] which creates an mp3 file with the spoken data from text. Finally, with pygame library [21] it is possible to play the created file. This flow is repeated continuously providing effective support to the user.

The examples of recognised commands and the respective responses are:

- *hello*: it says "hello"
- *how are you*: it says "I'm fine and you?"
- *follow me*: it says "Let's go" and starts walking forward
- *turn right*: it says "turn right" and turns right
- *turn left*: it says "turn left" and turns left
- *go back*: it says "ok I'll go backward" and go backward
- *play*: it plays random music

Api Manager This process is mainly responsible for providing external APIs. The APIs allows the user to access the PyBot resources from a different location without the need to have the PyBot nearby. The APIs handle different resources:

- *movements*: movements performed by the PyBot
- *stream*: camera stream of the PyBot

These APIs are developed using the Flask framework,

Fit Manager This process is mainly responsible for the gathering of data from the wristband. The process can acquire from the user: steps done, distances travelled, calorie burned, heart rates.

The process is based on the library provided by [22] with some updates, that exploits the BLE protocol to connect to the wristband.

Emotion Manager This process is mainly responsible for the recognition and handling of user emotions. The process captures every 10 seconds a frame from the camera module and then sends the image captured to the PyServer, which aims to recognise the emotion and send back the result. The way of emotion recognition by PyServer is described in the next subsection. The emotion manager when receives the result from the PyServer, analysis the result, and according to the emotion obtained it starts a conversation with the user.

3.3 PyServer

The PyServer is designed to act as the remote server of the system proposed. It is responsible for the storing of the information generated by both the MyFit

and the PyBot, as well as for the operations of emotion recognition.
As mentioned at the beginning of this section the architecture used is the REST API which can use an interchangeable language to manipulate the HTTP protocol and to provide resources.

The APIs handle different resources:

- *users*: users registered in the system
- *foods*: foods eaten by users
- *activities*: training activities done by users
- *steps*: steps done by users
- *distances*: distances travelled by users
- *calories*: calories burned by users
- *heart rates*: heart rates of the users

The PyServer is designed to be secure, provided an authentication strategy, mandatory to access resources, based on a token, JWT. The token allows identifying users and their role, to restrict the resources according to which actor is logged in.

To allow emotion recognition a pre-trained CNN, developed by Serengil [15], is used. The CNN is trained on the FER 2013 dataset [16]. Seven emotions can be recognised: anger, disgust, fear, happiness, sadness, surprise, neutral.

The flow of the activities to predict emotions is shown in Fig. 6.

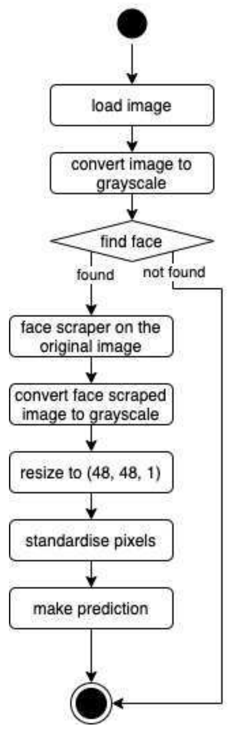


Fig. 6: PyServer: emotion recognition flow

When the emotion recognition resource is called the server expects to receive an image as input. When the image is loaded, the conversion of its colour scheme is applied, which is changed in grayscale. The segment where the face is located is obtained, if present, by using an OpenCV function [23]. If a face is found, the section where the face is located is cropped to create a new image with only the face. The new image is converted to grayscale, resized to a size of (48, 48, 1) to fit well in the machine learning model, and the pixels in the image are standardised, this means that all pixels that were initially in a range from 0 to 255 are now converted into a range from 0 to 1 to improve prediction performance. Now the image can be used as an input for the machine learning model to obtain the prediction.

The PyServer is developed in Python, the APIs are developed using the Flask framework. For the storage is used a relational database to provide a well-defined structure with multiple relationships.

4 Tests and Evaluation

To evaluate the functionality offered by the system, several usability tests had been defined to be submitted to different people to evaluate the ease of use of the system and the correct functioning, but due to the current critical world situation, caused by the spread of a disease on a large scale, the tests could not be completed.

Three macro groups of tests have been identified, one in which the MyFit app is tested, one in which the PyBot is tested, and one in which the facial detection features are tested. For each test, the user had to be asked to complete some activities to test the usability and correctness of the system. The experiments were conducted in the laboratory. Participants were given a set of tasks to test MyFit and PyBot. The following results were recorded:

- Time to complete a task per user
- Number and type of error per task
- Number of errors per unit time per user
- Number of users completing a task successfully

After carrying out the tests a questionnaire on how the users feel about using the product, by asking them to rate it along with a number of scales, after interacting with it.

The facial detection test was carried by one person only, due to the covid19. The person involved is a male of 23 years old. The test was set up in a room of around 10 sqm with no artificial light with few background noises. The person was asked to stand still in front of the PyBot for 4 minutes, two minutes looking at it and two minutes with their backs facing it. The test is repeated four times at different distances: 50 cm, 100 cm, 150 cm, 200 cm. The aims are to evaluate the performance of the underline face detection infrastructure when the user is in front of the robot in both looking and not looking scenarios.

The data obtained are classified into two classes, face recognised (1) and face not recognised (0), so it will be:

- True Positive (TP): the person is looking at the PyBot and the face is recognised
- False Positive (FP): the person is not looking at the PyBot and the face is recognised
- True Negative (TN): the person is not looking at the PyBot and the face is not recognised
- False Negative (FN): the person is looking at the PyBot and the face is not recognised

The confusion matrices from the result of each experiment are shown in table 1.

50 cm		
	not recognised	**recognised**
not present	14	1
present	0	15

100 cm		
	not recognised	**recognised**
not present	15	0
present	1	14

150 cm		
	not recognised	**recognised**
not present	15	0
present	0	15

200 cm		
	not recognised	**recognised**
not present	15	0
present	2	13

Table 1: Confusion matrices of face detection tests, obtained at different distances among the person and the PyBot: 50cm, 100cm, 150cm, 200cm

Distance	Average execution time (s)	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)
50 cm	3.07	96.67	100	93.75	100
100 cm	3.05	96.67	93.34	100	93.75
150 cm	3.02	100	100	100	100
200 cm	3.04	93.33	86.67	100	88.24

Table 2: Performance of face detection

Table 2 summarises the classification performance of face detection in different distances.

The evaluation metrics used are calculated as following:

Accuracy:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision:

$$PREC = \frac{TP}{TP + FP}$$

Sensitivity:

$$SN = \frac{TP}{TP + FN}$$

Specificity:

$$SP = \frac{TN}{TN + FP}$$

Results show that the underlying infrastructure is robust and can also be used for further emotion detection, which is not covered in this paper. However, this must be considered only as a preliminary study as the experiments were conducted on one person and the images collected are fairly few.

5 Discussion

The system aims at gathering the user's health data without destroying what is his daily life. Using different heterogeneous systems has allowed reaching a good compromise and it has allowed monitoring the user's activities even when the user decides to leave home. In addition, the use of a robotic system to communicate with the user allows the distance between the user and the system to be minimised making the user feel more comfortable.

To better understand what the advantages and disadvantages of this system are, it is good to analyse the three proposed sub-systems differently.

5.1 MyFit

The MyFit is responsible for gathering the major number of information from the user. After considerable and prolonged use, several advantages have emerged. The most important is that the user as soon as he accesses the application starts a background service that collects information from the wristband without the user having to interact with it. The only limitation of this service is that the app must be run in the background even if closed, otherwise, the service will not work until the app is reopened. Another advantage is that the app provides the user an all in one place to record his daily habits, usually a user has to install different applications. In addition, the user can check his trends, or even more useful can check the trends of a family member. The main issue raised up is the network reliability for the data storage, because there is an algorithm that sends the data to the server and stores in local the information added by the user, but in case of network connection leak, there is no way to understand if the data was been sent to the server or not and if so retry. MyFit can be connected to a wristband but is now only compatible with Xiaomi Mi Band 3, so it can think of expanding compatibility to many more wristbands.

5.2 PyBot

The PyBot is responsible for acquiring information from the user, but user interaction is even more useful. PyBot can offer several advantages to the user. The most important thing is that if the user is not happy to wear a wristband, the data can be acquired by the PyBot, leaving the user free of wearable devices, but at the same time under the control of the robot. It is also important that PyBot can be used to encourage the user, for example on a bad day, by providing support through conversation or playing music. During the development different design issues are raised:

- *Energy issue*: the PyBot has several sensors connected to itself, so the average energy consumption is significant. Due to its mobility, it cannot be charged for long and often, otherwise, it becomes useless. Optimising energy consumption is, therefore, the main question that must be solved
- *Network reliability*: the PyBot uses different services over the internet, the emotion recognition provided by the remote server, the text-to-speech and the speech-to-text provided by a third part. Therefore, is necessary to ensure a reliable connection or provide an algorithm that can compute in local.
- *Quality of the modules*: the PyBot integrates different modules, fig 4. The use and the reliability of these modules is a key concept to allow the PyBot to work well. Problems relating to modules are identified during tests. The camera used is not capable of operating in a low light environment and the resolution is poor, so it can lead to user emotion confusion due to low image quality in everyday use. The microphone used takes on a lot of ambient noise, so it can lead to wrong speech recognition. The distance sensor is not able to identify all the obstacles in front of it. The distance sensor utilises ultra-sonic waves that are useful because they are not influenced by object light, colour or transparency, but they are not reflected by soft materials so that the robot sometimes fails to identify the person in front of it.
- *Emotion recognition*: the emotion recognition is one of the most complex machine learning problem because the emotions of people can affect differently the face of each person, so it is very difficult to identify a pattern. To improve the reliability of emotion recognition is possible to integrate with face images also audio recording and analysis of body movements [24].

To solve the design-related issues, just assume the components are easily replaceable thanks to the system's modularity, which can lead to great improvements according to needs.

5.3 PyServer

The PyServer is the main source of storage. Since a huge amount of extensive data of the user is collected the privacy invasion will be very serious. After considerable and prolonged usage, it can be established that the system used for authentication, token, enables to avoid possible holes in the system while protecting user data. Furthermore, all data collected are stored following the GDPR guidelines. The other function of PyServer besides storage is to recognise emotions, the server-side emotion recognition can be modified to take advantage of new sources of knowledge according to the previous subsection on PyBot.

In summary, a good level of data collection has been achieved which can lead to different new scenarios. One possible scenario is to create a dataset containing the fitness activities, the foods eaten, the training activities, the emotions belong to the users, related to their health, in order to create a machine learning algorithm that can predict the health status of a user collecting only this information.

6 Conclusion and Future Works

A heterogeneous framework for controlling and improving health and well-being is proposed in this paper. A smartphone app, a remote server, a robot and a wristband make up the whole system. The mobile app facilitates the processing of information from the wristband, allowing, also, the user to record the foods consumed and the activities performed. The robot is capable of gathering information from the wristband without user intervention and is capable of understanding the user's emotional state to assist it, additionally the robot's abilities to talk and listen lead to reduce the gap between robot and human. A limitation of the study is the lack of a usability study, due to the closure of university campuses caused by the spread of Covid-19. In fact, to improve the performance obtained one of the most important future works will be to organise test sessions in a controlled environment to consolidate the work done.

In conclusion, it can be noticed that the mutual exclusivity between the app and the robot let the user a greater degree of freedom maintaining a good level of information collection and thanks to the use of the PyBot it is possible to make the user feel safer and more peaceful. Furthermore, the possibility of having a remote view of family members can be used in an easy way to monitor their habits without being too invasive. The proposed system represents an early step towards automated care in everyday life which opens the doors to many new scenarios. The system could be used as an encouragement for people who are reluctant to play sports or other physical activity, motivating them to increase their participation on days when they are more sedentary, to maintain their good health and to prevent obesity. The system could be used as a help for elderly people, who wish to maintain their autonomy but are required to seek third-party assistance.

Future work will be carried out to incorporate other wristbands with the MyFit app, to improve the facial detection and emotion detection algorithms, and to undertake a large scale of data collection and evaluate the system.

7 Acknowledgement

This research is partially supported by the Beitto-Ulster collaboration programme.

References

1. M. Weiser, "The computers for the 21st century," Sci. Amer., September 1991.
2. D. Saha, and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," in Computer, vol. 36, no. 3, pp. 25-31, March 2003.
3. M. Satyanarayanan, "Pervasive computing: vision and challenges," in IEEE Personal Communications, vol. 8, no. 4, pp. 10-17, Aug. 2001.
4. A. Sangwan, and P. Bhattacharya, "Wireless Body Sensor Networks: A Review," International Journal of Hybrid Information Technology, vol 8, no. 9, pp. 105-120, 2015
5. R. Dobrescu, D. Popescu, M. Dobrescu, and M. Nicolae, "Integration of WSN-based platform in a homecare monitoring system," International Conference on Communications & Information Technology (CIT'10), pp. 165-170, July 2010.
6. H. Pei-Cheng, and W. Chung, "A comprehensive ubiquitous healthcare solution on an Android^{TM} mobile device," Sensors (Basel, Switzerland), vol. 11, June 2011.
7. World Health Organization, "Ageing and health," February 2018. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health.
8. G. Wilson, C. Pereyda, N. Raghunath, G. De La Cruz, S. Goel, S. Nesaei, B. Minor, M. Schmitter-Edgecombe, M. E. Taylor, D. J. Cook, "Robot-enabled support of daily activities in smart home environments," Cognitive Systems Research, 54, pp. 258-272, October 2018.
9. M.J. Mataric, "Socially assistive robotics: Human augmentation versus automation", Science Robotics, pp. 1-3, 2017.
10. J. Huang, W. Xu, S. Mohammed, Z. Shu, "Posture estimation and human support using wearable sensors and walking-aid robot," Robotics and Autonomous Systems, vol. 73, pp. 24-43, 2015
11. M. Goršič, R. Kamnik, L. Ambrožič, N. Vitiello, D. Lefeber, G. Pasquini, and M. Munih, "Online Phase Detection Using Wearable Sensors for Walking with a Robotic Prosthesis," Sensors, vol 14, pp. 2776-2794, 2014
12. M. Chen, Y. Ma, S. Ullah, W. Cai, E. Song, "ROCHAS: Robotics and Cloud-assisted Healthcare System for Empty Nester," 8th International Conference on Body Area Networks, BodyNets, pp. 217-220, September 2013
13. Y. Ma, Y. Zhang, J. Wan, D. Zhang, N. Pan, "Robot and cloud-assisted multi modal healthcare system", Cluster Comput 18, 1295–1306, May 2015

14. Wikipedia contributors, "Json," April 2020, [Online]. Available: https://en.wikipedia.org/wiki/JSON
15. S. I. Serengil, "TensorFlow 101: Introduction to Deep Learning for Python Within TensorFlow". [Online]. Available: https://github.com/serengil/tensorflow-101
16. "Challenges in Representation Learning: Facial Expression Recognition Challenge," 2013. [Online]. Available: https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/overview
17. R. Sridhar, H. Wang, P. McAllister, H. Zheng, "E-Bot: A Facial Recognition Based Human Robot Emotion Detection System," Proceedings of the 32nd International BCS Human Computer Interaction Conference (HCI), July 2018
18. "SpeechRecognition 3.8.1," [Online]. Available: https://pypi.org/project/SpeechRecognition/
19. "googletrans 2.4.0," [Online]. Available: https://pypi.org/project/googletrans/
20. "gTTS (Google Text-to-Speech)," [Online]. Available: https://pypi.org/project/gTTS/
21. "pygame 1.9.6," [Online]. Available: https://pypi.org/project/pygame/
22. Y. Ojha, "MiBand3," [Online]. Available: https://github.com/yogeshojha/MiBand3
23. "opencv-python 4.2.0.34," [Online]. Available: https://pypi.org/project/opencv-python/
24. M. El Ayadi, M. S. Kamel, F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases,", Pattern Recognition, Volume 44, Issue 3, pp. 572-587, 2011,