Mesh Network of eHealth Intelligent Agents in Smart City: A Case Study on Assistive Devices for Visually Impaired People

Dmytro Zubov[1][0000-0002-6807-0924], Utku Kose[2][0000-0002-9652-6415],

Ali Jasim Ramadhan[3][0000-0003-3253-3525], and Andrey Kupin[4][0000-0001-7569-1721]

[1] University of Information Science and Technology, Ohrid 6000, Macedonia
dzubov@ieee.org
[2] Suleyman Demirel University, Isparta 32260, Turkey
utkukose@sdu.edu.tr
[3] University of AlKafeel, Kufa 31003, Iraq
ali.j.r@ieee.org
[4] Kryvyi Rih National University, Kryvyi Rih 50027, Ukraine
kupin.andrew@gmail.com

Abstract. Over 253 million people across the world today are estimated to be blind or visually impaired. White canes and guide dogs remain the most preferred methods of aid despite the availability of smart technologies and over a hundred assistive electronic devices developed within the last three decades. The main sticking points are unaffordability, implementation of different technologies in one device, and integration with the existing network. The developed assistive device is designed to overcome these obstacles. It is low-cost and priced competitively at USD 70. It is based on the palm-sized computer Raspberry Pi 3 B with Pi camera and ultrasonic sensor HC-SR04, and it features MQTT IoT protocol allowing it to communicate with other intelligent eHealth agents. The basic functionalities of the device are measurement of the distance to the nearest obstacle using the detector HC-SR04 and recognition of human faces by the OpenCV and face recognition module of Adam Geitgey. Objects and places around the B&VI are indicated by the Bluetooth devices of classes 1-3 and iBeacons. Intelligent eHealth agents cooperate with one another to efficiently route data from/to clients in the smart city mesh network via MQTT and BLE protocols. The presented soft-/hardware was successfully tested and accorded a score of 95.5 %.

Keywords: mesh network, eHealth intelligent agent, blind and visually impaired, smart city, Raspberry Pi 3 B, HC-SR04, OpenCV.

1 Introduction

Nowadays, around 30 million blind and visually impaired (B&VI) Europeans need assistance for the spatial cognition indoor and outdoor [1]. World Health Organization showed in 2017 that over 253 million people are estimated to be B&VI, over 19 million children of age below 15 have serious problems with vision worldwide [2]. The white canes and guide dogs remain the common aid associated with this disability in spite of over a hundred different assistive electronic devices developed last three decades [3, 4]. The additional training of B&VI, especially children [5], to use the innovative hardware and its integration into the existing networks are the main difficulties nowadays. The first problem is solved by short-term courses, where B&VI people learn how to handle the assistive devices. The second problem does not have a single solution due to the heterogeneous soft-/hardware, as well as the standard smart city infrastructure, which is a dominant platform for communication of assistive devices, was not developed yet.

Internet of Things (IoT) hardware is produced by many companies such as Raspberry Pi Foundation, Arduino, Bosch, Silicon Labs. IoT software is represented by different operating systems like Raspbian, Android Things, Windows 10 IoT, Ubuntu Core. The assistive devices exchange data within metropolitan area networks [6] via IoT data protocols like MQTT and CoAP [7] based on different network technologies such as WiFi, Bluetooth, Ethernet [8]. On a short distance of up to 10 m, most IoT hardware, smartphones, and tablets implement the Bluetooth classes 2 or 3 to be connected directly [9]. Standard iBeacons [10, 11] have an approximate range of 70 meters for the peer-to-peer Bluetooth low energy (BLE) communication. For longer distances, the client-server and/or wireless mesh network architectures [12] are applied.

For the spatial cognition, B&VI people require specific functionality from the assistive devices:

1. Geographic Information and Global Positioning Systems (GIS and GPS) for the navigation.
2. Ultrasonic/optical detection of the obstacles and distances to them.
3. Computer vision detection and recognition of the people and other objects like animals in front of and/or around the B&VI.
4. Indication of objects and places around B&VI via the BLE techniques, e.g. RSSI, AoA, and ToF, and recently proposed BLE based protocols such as iBeacons (Apple Inc.) and Eddystone (Google Inc.) [13].
5. Integration into existing smart city infrastructure using wireless technologies such as Bluetooth and WiFi.
6. Other navigation assistance to solve the last 10 meters problem, e.g. the sound marking of the objects like the entrances and exits.

For the time being, none of the existing B&VI assistive devices implements the above-stated functionality (1)-(6) jointly. The most promising approach is the smart city mesh network of eHealth intelligent agents since the number of network-connected devices is estimated to be 6.58 per person around the world in 2020 [14]. It is a huge increase comparing with 2003 (0.08), 2010 (1.84), and 2015 (3.47), which reflects the global tendency – IoT becomes the Internet of Everything, i.e. almost every new smart device has unique network ID. BLE is preferable technology to connect the nodes directly (i.e. peer-to-peer [15]), dynamically, and non-hierarchically within the mesh network on the short distances of up to 90 m due to the energy efficiency. WiFi is commonly used to connect the nodes for longer distances in the smart city networks [16, 17]. The low-cost iBeacon mesh network might be developed with Raspberry Pi, Bluetooth module HM-10, and Python library Bluepy for interfacing with BLE devices through

Bluez (the official Linux Bluetooth protocol stack [18]). However, the prices of assistive devices for B&VI with advanced features such as the computer vision and iBeacon object identification [19] are high. For instance, the price of OrCam MyEye wearable assistive device is USD 3,500 [20]; it detects what the B&VI is looking at, recognizes the presence of text, and speaks the text aloud.

The goal of this paper is to present the developed assistive device for the B&VI spatial cognition with above-stated functionality (2)-(5) and affordable price of USD 70. The emphasis is on the soft-/hardware and its integration into the mesh network of eHealth intelligent agents for the B&VI using the Python programs on the Raspbian platform.

This paper is organized as follows: In Section 2, the related work is presented. In Section 3, the mesh network of eHealth intelligent agents is suggested as the B&VI assistive subsystem in smart city infrastructure. In Section 4, the eHealth intelligent agent for the B&VI is discussed through its components: the ultrasonic sensing of the distances to obstacles through the range sensor HC-SR04; the computer vision detection and recognition of human faces in front of the B&VI via the Raspberry Pi camera, OpenCV library, and face recognition module of Adam Geitgey; the indication of objects and places around the B&VI using the Bluetooth devices of classes 1-3 and iBeacons. Conclusions are summarized in Section 5.

2 Related Work

Previous studies on the mesh network of eHealth intelligent agents for B&VI were mostly published in the Artificial Intelligence (AI) and IoT journals and conference proceedings since assistive devices are based on the intelligent algorithms and IoT soft-/hardware. They can be classified by the above-stated functionality (1)-(6).

In [21], the GIS acquires information from the obstacle avoidance sensors, positioning devices, and the end-user, and then references these to the spatial database. This spatial information is conveyed to the end-user via audio- and/or tactile-based interfaces a few steps forward. In [22], the wearable smart navigation system is presented using GPS and ultrasonic sensor. A solar panel provides the additional power to the system. Microsoft Australia has launched the Soundscape app that uses audio beacons and AI technology to provide navigational information for B&VI [23]. However, the image processing is not applied in [21-23]. Nowadays, the B&VI have explicit navigation instructions provided by ordinary smartphones with GPS.

In [24], the spatial cognition of B&VI is based on the measurement of the distance to an obstacle using ultrasonic sensor HC-SR04. The information is pronounced to the B&VI through headphone and MP3 player with Arduino Uno. Assistive devices were successfully tested at the Instituto para Ciegos y Débiles Visuales “Ezequiel Hernández Romo”, San Luis Potosi, Mexico.

There are different types of the computer vision detection and recognition algorithms [25-28], e.g. Eigenfaces, Fisherfaces, Histogram of Oriented Gradients (HOG), Local Binary Patterns Histograms, Scale Invariant Feature Transform, Speed Up Robust Features, Viola-Jones Face Detection. These methods extract the image features and perform the matching with the input image. Some of them like HOG might be used for the detection and recognition of different categories of objects, e.g. humans and animals. The support vector machine classifier with the HOG feature extractor is the most popular solution for the object detection and classification [29]. In [30], a comparison of the sample linear discriminant function (LDF) and Euclidean distance classifier (EDC) showed that the sample EDC outperforms the sample LDF if the number of features is large relative to the training sample size. Nowadays, different computer vision libraries, e.g. OpenCV, Vlfeat, PCL, and SimpleCV [31-33], use the above-stated methods to analyze images via local features extraction and matching. In general, the best method is specified by the problem.

In [34], the depth-based algorithm finds the candidate traversable directions using the depth image. The proposed approach assumes that the nearest obstacle is always at the bottom of the depth image, and it selects only a line at the bottom of the image as input. The ultrasonic sensor is applied as well. The object recognition is not discussed in [34]. The wearable assistive device is developed with the RGBD camera and FPGA CPU in [35]. The semantic categorization of detected obstacles is based on the convolutional neural network. The experiments showed good detection performance close to 98 %, although the object categorization is less than 72 % of correctness. The sensed area for obstacle detection is from 0.5 m to 3 m only, and the price is not presented in [35].

In [10], iBeacons are proposed for the spatial marking in the low-cost and energy-effective IoT navigation system for the B&VI. The Raspberry Pi board detects the obstacles with the ultrasonic sensor. The information is provided to the end-user via the Bluetooth headphones.

There is no commonly accepted definition for the smart city [36, 37]. A concept of the inclusive smart infrastructure [38], where physical and digital barriers are eliminated, targets the needs of the B&VI. This idea is implemented through the connected and/or mesh networks [17, 18] with eHealth intelligent agents [39, 40], which communicate via IoT data protocols such as MQTT, CoAP, XMPP, DDS, AMQP [7, 16]. MQTT IoT software can be executed on thin clients like Arduino Uno since it takes approximately 10 KB of the random access memory. It was shown that MQTT brokers work reliably with 100,000 publishers and 100 subscribers [41] that satisfies the requirements to smart city networks.

In [24], the assistive device supports the golf game of B&VI with sound marking of golf flagsticks with Wi-Fi NodeMcu Lua ESP8266 ESP-12 boards and active buzzers, which are remotely controlled by a person with good vision via Intranet HTML websites. The same concept go-to-sound G2S is used in the assistive system for the B&VI [42] developed for Ukrainian Association of the Blind.

In [4], twenty five different assistive devices are compared using the following characteristics: real-time / not real-time mode, coverage (indoor, outdoor, both), time (day, night, both), range (less than 1 m, 1 m to 5 m, over 5 m), object type (static, dynamic, both). 72 % of the presented products have at least three characteristics that are not fully satisfied, e.g. the device Eye Subs provides outdoor coverage but not the indoor one, the detection range is less than 5 m due to the ultrasonic limitation, and it detects only the static but not dynamic objects. The prices are not shown. The authors pointed out that the ideal device has to recognize the surrounding environment at all times and everywhere.

The practical use of the B&VI assistive devices shows the following outcomes for the soft-/hardware:

- GIS and GPS are the standard components of the smartphones soft-/hardware nowadays.
- The ultrasonic detectors are more preferable to the optical one for the wearable hardware because of the price and operational distance; for instance, an ultrasonic sensor HC-SR04 (price USD 1.5 approx.) measures distance of up to 5 m and an infrared proximity sensor GP2Y0A21 (price USD 3.5 approx.) – up to 0.8 m only.
- Developers prefer the Bluetooth transmitters/receivers based on the standard BLE protocols such as iBeacon. Devices are connected by Bluetooth on a short distance of up to 90 m, by WiFi – longer distances within the local area network of the smart city.
- Smartphones are used for general-purpose tasks such as face detection/recognition during daytime and GPS navigation. IoT electronics like Raspberry Pi 3 B [43] and Arduino Tian [44] are applied for the specific solutions, e.g. night vision and ultrasonic sensing, since additional sensors are utilized.
- Python is the leading programming language for the implementation of the intelligent algorithms on Raspbian.

The outcomes for the assistive methods are as follows:

- Standard libraries are commonly used for face recognition, e.g. OpenCV with Python on the palm-sized computer Raspberry Pi 3 B [45].
- The sound marking is turned on only if the B&VI is next to the target object since the continuous sound bothers the people with normal vision.
- The B&VI need only specific information about surrounding objects because of their diversity and large quantities.

Analysis of the above-stated work shows that the spatial cognition is based on the GPS navigation via smartphones, the ultrasonic sensing of the obstacles, computer vision detection / recognition / categorization of the surrounding objects, and BLE technology for the spatial marking. None of the known assistive devices for the B&VI combines the above-stated functionality (1)-(6), they are not the standard components of the smart city infrastructure, as well as they do not implement the mesh networking for the communication with other IoT equipment.

3 Mesh Network of eHealth Intelligent Agents for the B&VI: A Methodology Outline

The initial idea was to develop the mesh network of eHealth intelligent agents for the B&VI, where the infrastructure nodes could be connected directly/indirectly, dynamically/statically, and hierarchically/non-hierarchically and cooperate with one another to efficiently route data from/to clients. The functionality (2)-(5)

We use the meet-in-the-middle methodology [46] to outline the mesh network of eHealth intelligent agents for the B&VI. The proposed smart city inclusive component with the mesh network of eHealth intelligent agents for B&VI is presented in Fig. 1. The data-driven assistive approach is a core of this component. Here, MQTT IoT protocol connects the eHealth intelligent agents, as well as other parts of smart city (smart buildings, smart mobility, smart energy, smart water, smart health, smart waste management, smart industry, smart infrastructure development and monitoring, etc.) communicate with one another via IoT protocol(s) [7]. The Python programs, addresses of MQTT brokers and Bluetooth devices, MQTT passwords, MQTT topics, recognition rules, MP3 files, and other relevant information are downloaded from the file server(s) to the eHealth agents when they start or by request. If the Internet is not connected, previously downloaded/installed files are loaded from the SD card. In addition, the eHealth devices might transmit/receive the information by Bluetooth, e.g. classes 1-3 and iBeacons.

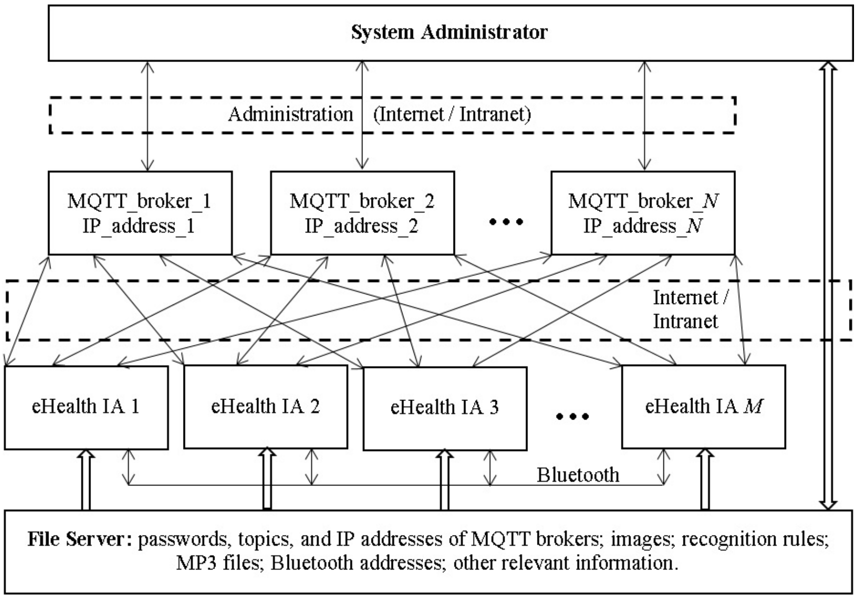

Fig. 1. A smart city inclusive component with mesh network of eHealth intelligent agents for B&VI.

Raspberry Pi 3 B boards can handle all three MQTT roles – broker, publisher, and subscriber [47]. For this purpose, the following software is installed:

- The open source MQTT message broker Mosquitto to start the MQTT broker, publisher, and subscriber on Raspberry Pi 3 B through the main repository:

```
sudo apt-get install -y mosquitto mosquitto-clients
```

- The Python library paho-mqtt for implementing the publisher and subscriber in the Python 3 program:

```
sudo pip3 install paho-mqtt
```

The presented in [47] Python code can be replicated with small corrections such as MQTT topic and IP address of MQTT broker.

In the proposed hardware configuration, the Raspberry Pi 3 B is a basic microcomputer. The prototype and the final product are presented in Fig. 2. The Raspberry Pi 3 B+ board was considered as well, but it is too sensitive to the power supply. In particular, the warning "under-voltage detected" is displayed if Raspberry Pi 3 B+ is supplied by the mobile power banks; two power banks have been applied, and microcomputer restarted every two minutes. The same external batteries are working well with Raspberry Pi 3 B board.

In this project, Arduino boards are used for simple operations like the data acquisition, e.g. the heart rate measurement with sensor KY-039, Arduino Uno, and Ethernet Shield [48-50] (see Fig. 3).

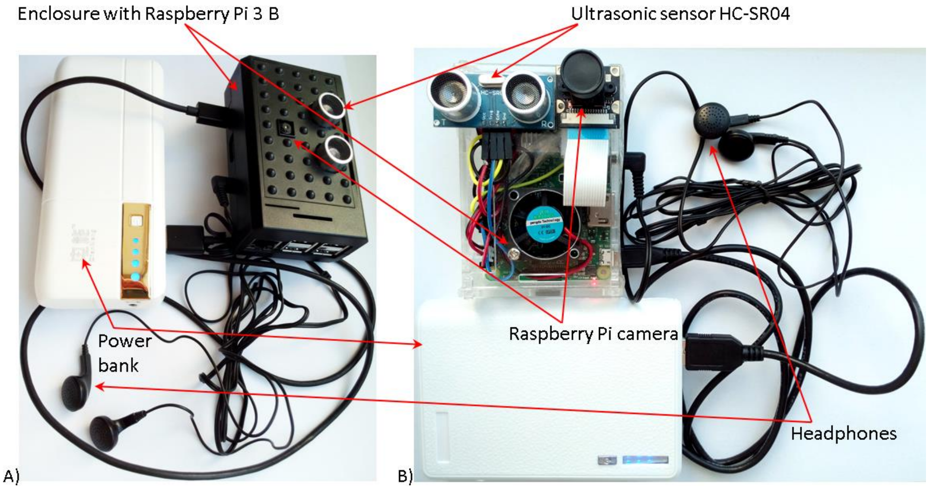

Fig. 2. The final product (A) and prototype (B) of the B&VI assistive device: Raspberry Pi 3 B board with the camera, ultrasonic sensor, headphones, and mobile power bank.

Fig. 3. A prototype of the eHealth assistive device for the heartbeat measurement: Arduino Uno board with Ethernet shield and module KY-039.

The basic functions of the B&VI assistive device are as follows: ultrasonic measurement of the distance of up to 5 m to the nearest obstacle via the range detector HC-SR04; computer vision detection and recognition of the human faces in front of the B&VI based on the Raspberry Pi camera; indication of the objects and places around B&VI using the Bluetooth devices of classes 1-3 and iBeacons; integration of the assistive device into existing smart city infrastructure through the MQTT IoT data protocol.

Four Python programs were developed to implement the above-stated tasks. They start in parallel when the assistive device is turning on. For this purpose, the launch script is developed as follows [51]:

1st step. The script file launcher.sh is developed in the text editor as follows:

```
cd /home/pi/Documents/Download
sudo python Download1.py
7z x Files.zip -y
sudo python3 pi_face_recognition.py --cascade haarcascade_frontalface_default.xml --encodings encodings.pickle & sudo python3 Ultra4.py & sudo python3 Bluetooth4.py & sudo python3 MQTT1.py
```

In the first line of file launcher.sh, the current directory is changed to the specified "/home/pi/DIP/Experiment/Download". This directory includes all project files.

In the second line of file launcher.sh, the Python program Download1.py is started. This program downloads zip-file Files.zip from the Internet file server. The URL is unique for every assistive device to satisfy the requirements of the B&VI person. The Python code is as follows:

```
import requests
url = 'http://URL_is_here'
fileName = '/home/pi/Documents/Download/Files.zip'
req = requests.get(url)
file = open(fileName, 'wb')
for chunk in req.iter_content(100000):
  file.write(chunk)
file.close()
```

In the third line of file launcher.sh, zip-file Files.zip is unzipped by program 7z. Command "x" extracts all files from the archive Files.zip to the current directory. Switch "-y" assumes YES on all queries.

In the fourth line of file launcher.sh, four Python programs start in parallel: pi_face_recognition.py (detection and recognition of the human face), Ultra4.py (ultrasonic measurement of the distance to nearest obstacle), Bluetooth4.py (an indication of the Bluetooth devices of classes 1-3 and iBeacons), MQTT1.py (communication with other intelligent agents via the MQTT IoT data protocol). These four Python programs are discussed below.

2nd step. Make the script file launcher.sh executable:

```
chmod 755 launcher.sh
```

3rd step. Add the logs directory for records of any errors in crontab, which is a background (daemon) process that executes scripts at specific times (at the start in our case):

```
mkdir logs
```

4th step. Add the script launcher.sh to crontab. The following command is executed in terminal:

```
sudo crontab –e
```

Then, the following code is added at the bottom of the text file:

```
@reboot sh /home/pi/Documents/Download/launcher.sh /home/pi/Documents/Download/logs/cronlog 2>&1
```

The information is pronounced to the B&VI through headphones. All phrases are stored in MP3 files, and hence the language localization can easily be done by the simple replacement of the MP3 files on the file server in the end-user specific folder. In the Python program, the command-line MP3 player mpg321 is implemented as follows (MP3 file DeviceIsStarting.mp3 is an example):

```
os.system("mpg321 DeviceIsStarting.mp3")
```

To start MP3 files sequentially, the following Python code is applied (psutil is a cross-platform library for retrieving information on running processes and system utilization):

```
flag=1 # This variable is the semaphore: 1 – another MP3 file is playing
# In the while loop, we wait while another mpg321 player is stopped, i.e. flag=0
while flag==1:
    flag=0
    try:
        for pid in psutil.pids():
            p = psutil.Process(pid)
            if p.name()=="mpg321":
                flag=1
    except:
        flag=1
        time.sleep(2)
        print("We have an error in PIDs")
```

The presented approach of the microcomputer Raspberry Pi 3 B has the following drawbacks:

- In spite of the low cost of the assistive device (USD 70 approx.), the price is high for some B&VI. The solution is to develop the mobile application(s) for Android and iOS using Xamarin cross-platform technology [52]. In this case, the B&VI will use their smartphones to download and install software at no cost.

- The B&VI listen surrounding environment to identify objects. At the same time, they use headphone(s) to acquire audio information from smartphones and other devices. If they have different equipment, the audio mixer is additionally needed to manage audio from each service.

4 eHealth Intelligent Agent for the B&VI: Soft- and Hardware Implementation

4.1 Ultrasonic Measurement of the Distance to the Nearest Obstacle

The initial idea was to develop the low-cost and reliable assistive wearable hardware to detect the obstacles in front of the B&VI and to measure the distances to them.

The hardware is based on the ultrasonic range sensor HC-SR04, which works well on the distances of up to 5 m [53]. Since the device is a small-sized, the B&VI can easily rotate it. The Raspberry Pi 3 B GPIO pins are of maximum voltage 3.3 V and the power supply is 5 V. Hence, the voltage divider with resistors 1 kOhm and 2 kOhm is applied to the detector Echo output. The Python code for acquiring the information from the sensor HC-SR04 is presented in [53]. For the user-friendly representation of distances, the range is split into five intervals – less than 100 m, 200 cm, 300 cm, 400 cm, and 500 cm. This information is repeated every two minutes if it is not changed. The fragment of Python program is as follows:

```
if (distance<=100 and distance>0.5):
os.system("mpg321 LessThanOneMeter.mp3")
else:
    if (distance<=200):
        os.system("mpg321 LessThanTwoMeters.mp3")
    else:
        if (distance<=300):
            os.system("mpg321 LessThanThreeMeters.mp3")
        else:
            if (distance<=400):
                os.system("mpg321 LessThanFourMeters.mp3")
            else:
                if (distance<=500):
                    os.system("mpg321 LessThanFiveMeters.mp3")
                else:
                    os.system("mpg321 DistanceUnidentified.mp3")
```

The ultrasonic sensor has the following drawbacks [54]: it cannot detect objects, which distract sound due to the complex surface texture or too large angle between sensor and object; the sound could hit another surface before returning to the sensor instead of bouncing back from the target directly (the ghost echo) causing miscalculated timing; the temperature and sound noise interference.

Detection of the obstacle in front of the B&V takes approximately 0.3 sec, which is quite acceptable for the slowly walking B&VI.

If the calculated value of the distance is greater than 500 cm or less than 0, the end-user receives a message that distance is unidentified.

The most likely future improvement is the use of a range sensor with better measurement accuracy and/or detector based on another principle, e.g. the optical one.

4.2 Computer Vision Detection and Recognition of the Human Faces in Front of the B&VI

The initial idea was to develop the low-cost and reliable assistive wearable hardware to detect and recognize the known faces and number of the unknown one.

The hardware is based on the Raspberry Pi camera [45] and computer vision library OpenCV [31] without the Python virtual environment [32]. It was found that OpenCV is not working properly from the script file at the Raspberry Pi 3 B startup if it is installed in the Python virtual environment. The face recognition project is developed as follows:

1st step. Install OpenCV 3.2 Python/C++ on Raspberry Pi 3 B [31].

2nd step. Install Davis King’s dlib toolkit software [45].

3rd step. Install Adam Geitgey’s face recognition module [45].

4th step. Install imutils package of convenient functions [45].

The project consists of the following files and directories:

- 31 MP3 files with audio information on the device state, spatial cognition, and the number of unknown faces: DeviceIsReady.mp3, InACenter.mp3, OnTheRight.mp3, OnTheLeft.mp3, NoUnidentifiedFaces.mp3, OneUnidentifiedFace.mp3, TwoUnidentifiedFaces.mp3, …, NineteenUnidentifiedFaces.mp3, TwentyUnidentifiedFaces.mp3, OverTwentyUnidentifiedFaces.mp3. The localization is based on the replacement of these files with the audio information in specified language, e.g. Chinese, English, Japanese, Macedonian, Ukrainian, Russian, Spanish.
- MP3 files with names of the people, which faces are detected and recognized.
- A directory dataset/ contains sub-directories with .jpg photos for each person that the facial recognition system detects and recognizes.
- A Python file encode_faces.py finds faces in the directory dataset/ and encode them into 128-D vectors [45, 55]. It is not changed in this project.
- A file encodings.pickle contains the face encodings (128-D vectors, one for each face).
- The OpenCV’s pre-trained Haar cascade file haarcascade_frontalface_default.xml is used to detect and localize faces in frames. It is not changed in this project.
- A Python file pi_face_recognition.py is the main execution script.

The Python program pi_face_recognition.py is based on the code presented in [45]. The corrections are as follows: playing MP3 files with audio information on the device state, spatial cognition, and the number of unknown faces; the code for displaying the image on a screen and drawing of predicted face name on the image was deleted to speed up the program.

It was found that the accuracy of face detection and recognition is better when the knowledge base contains more photos of different people [56]. Initially, the directory dataset/ contains two sub-directories with photos of two developers – Dmytro Zubov and Iryna Zubova (see Fig. 4; the images are scaled to a fixed height of 2000 pixels keeping their original aspect ratio).

Fig. 4. Two initial sets of photos with faces in the recognition knowledge base.

The drawback of the presented approach is an impossibility to estimate the accuracy of the face recognition. Hence, the most likely future improvement is the implementation of the new algorithm [56] for face detection and recognition with a calculation of the correct result probability.

4.3 Indication of Objects and Places Around B&VI via the Bluetooth Devices of Classes 1-3 and iBeacons

The initial idea was to develop the low-cost and reliable assistive wearable hardware to indicate the objects and places around the B&VI via the Bluetooth devices of classes 1-3, e.g. smartphones and Raspberry Pi boards, and BLE equipment, e.g. iBeacons.

To indicate the Bluetooth devices of classes 1-3 around the B&VI through the Python library bluetooth, a package libbluetooth-dev is installed as follows:

```
sudo apt-get install python-dev libbluetooth-dev
```

The following Python program shows the Bluetooth devices of classes 1-3 [57]:

```
import bluetooth
nearby_devices = bluetooth.discover_devices(lookup_names = True)
print ("found %d devices" % len(nearby_devices))
for name, addr in nearby_devices:
    print("%s - %s" % (addr, name))
```

To indicate the iBeacons around the B&VI, the Python code was developed similar to the presented in [18]. In addition, the bluepy library was installed:

```
sudo pip3 install bluepy
```

The difference of developed Python code with presented in [18] and [57] is as follows: the Audio output based on MP3 files and mpg321 player; the comparison of the indicated Bluetooth devices' addresses with the database of known objects and places – if the coincidence is found, the information is pronounced to the B&VI.

The drawback of the presented approach is the last 10 meters problem – the B&VI cannot identify the direction where the Bluetooth device is. The most likely solution to this problem is the RSSI based localization with three iBeacons [13] or fingerprinting [58].

4.4 Evaluation of Smart City Inclusive Component with Mesh Network of eHealth Intelligent Agents for B&VI

We use the criteria driven evaluation of the smart city inclusive component with mesh network of eHealth intelligent agents for B&VI. The usability [59] is applied as the quality characteristic with a score ranging from 0 (bad) to 10 (excellent) for five subsystems: 1 – ultrasonic detection of the distance to obstacle, 2 – face recognition, 3 – indication of objects via the smartphone Bluetooth, 4 – identification of the transport through iBeacon, 5 – communication with other smart devices based on MQTT IoT protocol. The experimental testbed consists of a public square, a public transport line, and two shopping centers in the Vishenka area, Vinnitsa city, Ukraine. The Bytereal iBeacon was installed in the public tram to identify the transport. Two smartphones were utilized to identify the people. Five participants, two blinds and three with normal sight, tested the assistive device and its communication with other equipment of smart city. The evaluation results are presented in Table 1.

Table 1. The evaluation results of the B&VI assistive device.

Testee (person)	Vision	Subsystem (score)
1	Normal sight	1 (10), 2 (9), 3 (10), 4 (10), 5 (0)
2	Normal sight	1 (10), 2 (9), 3 (10), 4 (10), 5 (0)
3	Normal sight	1 (10), 2 (9), 3 (10), 4 (10), 5 (0)
4	Blind	1 (9), 2 (9), 3 (9), 4 (10), 5 (0)
5	Blind	1 (9), 2 (9), 3 (9), 4 (10), 5 (0)

The mean values are as follows: 1 (9.6), 2 (9), 3 (9.6), 4 (10), 5 (0). The 5th subsystem usability equals zero since the outdoor network of MQTT clients and brokers are not installed in Vinnitsa yet. The average score is 9.55 for subsystems (1)-(4).

The final recommendations are as follows:

- develop the mobile version of the assistive device;
- improve the connection of power cable;
- replace the headphone to provide the audibility of the surrounding environment and assistive device information both.

5 Conclusion and Further Discussion

In this paper, the mesh network of eHealth intelligent agents for the B&VI is pre-sented as the smart city inclusive component. The low-cost and reliable wearable hardware priced at USD 70 was developed to assist the B&VI. The main constituents are as follows:

- Hardware: microcomputer Raspberry Pi 3 B with the ultrasonic range sensor HC-SR04, camera, headphone(s), mobile power bank.
- MQTT IoT protocol for the connection of eHealth agents and other components of the smart city. The Python programs, addresses of MQTT brokers and Bluetooth devices, MQTT passwords / topics, recognition rules, MP3 files, and other relevant information are downloaded from the file server(s) to the eHealth agents when they start.
- Ultrasonic measurement of the distance to the nearest obstacle via the range sensor HC-SR04.
- Computer vision detection and recognition of human faces in front of the B&VI using the Raspberry Pi camera, OpenCV library, and face recognition module of Adam Geitgey.
- Indication of objects and places around the B&VI through the Bluetooth devices of classes 1-3 and iBeacons.

All information is conveyed to the B&VI aurally via the headphone(s). For this purpose, the command-line MP3 player mpg321 is used in the Python program. Hence, the product localization can be easily done through the simple replacement of the MP3 files with information in the target language. Additional analysis of the transverse audio messages is applied to avoid simultaneous talking.

The presented mesh network of eHealth intelligent agents for the B&VI was successfully tested with a score of 95.5 % in Vinnitsa city, Ukraine. However, several drawbacks were found. The first was the ultrasonic sensor HC-SR04 sometimes failed to correctly detect objects due to distorted sound, ghost echo, temperature, and noise interference. Secondly, it was difficult to estimate the accuracy of the face recognition based on the Adam Geitgey’s module.

Besides the proposed mesh network of eHealth intelligent agents for the B&VI in this paper, other assistive technologies based on the Android/iOS mobile apps can be utilized for the B&VI spatial cognition. Moreover, in order to minimize the price and spread worldwide the achieved experience, we will develop mobile apps in the future work.

References

1. European Commission announces a €3 million Horizon Prize to develop a tactile display for the visually impaired, https://ec.europa.eu/digital-single-market/en/news/european-commission-announces-eu3-million-horizon-prize-develop-tactile-display-visually, last accessed 2018/10/29.
2. World Health Organization: Blindness and Visual Impairment, http://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, last accessed 2018/10/29.

3. Chaudary B., Paajala I., Keino E., Pulli P.: Tele-guidance Based Navigation System for the Visually Impaired and Blind Persons. In: Giokas, K., Bokor, L., Hopfgartner, F. (eds.) eHealth 360°, LNICST, vol. 181, pp. 9-16. Springer, Cham (2017).
4. Elmannai, W., Elleithy, K.: Sensor-Based Assistive Devices for Visually-Impaired People: Current Status, Challenges, and Future Directions. Sensors 17(3), art. no. 565 (2017).
5. Yaqoob, I., Hashem, I., Mehmood, Y., Gani, A., Mokhtar, S., Guizani, S.: Enabling Communication Technologies for Smart Cities. IEEE Communications Magazine 55(1), 112–120 (2017).
6. Dizdarevic, J., Carpio, F., Jukan, A., Masip-Bruin, X.: Survey of Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration, https://arxiv.org/abs/1804.01747, last accessed 2018/10/29.
7. Rackley, S.: Wireless Networking Technology: From Principles to Successful Implementation. Newnes, Oxford (2007).
8. Woodford, C.: Bluetooth, https://www.explainthatstuff.com/howbluetoothworks.html, last accessed 2018/10/29.
9. Trent, M., Abdelgawad, A., Yelamarthi, K.: A Smart Wearable Navigation System for Visually Impaired. In: Gaggi, O., Manzoni, P., Palazzi, C., Bujari, A., Marquez-Barja, J. (eds.) GOODTECHS 2016, LNICST, vol. 195, pp. 333-341. Springer, Cham (2017).
10. Shovic, J.C.: Raspberry Pi IoT Projects: Prototyping Experiments for Makers. Apress, New York (2016).
11. Zarzycki, A.: Adaptive Designs with Distributed Intelligent Systems – Building Design Applications. In: Herneoja, A., Österlund, T., Markkanen, P. (eds.), 34th eCAADe Conference Proceedings, vol. 1, pp. 681-690. University of Oulu, Finland (2016).
12. Zafari, F., Gkelias, A., Leung, K.K.: A Survey of Indoor Localization Systems and Technologies, https://arxiv.org/abs/1709.01015, last accessed 2018/10/29.
13. Rosebrock, A. (2018). Raspberry Pi Face Recognition, https://www.pyimagesearch.com/2018/06/25/raspberry-pi-face-recognition/, last accessed 2018/10/29.
14. Patnaik, P., Dinkar, R.: A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board. International Journal of Advanced Research in Computer Science 8(5), 2350-2352 (2017).
15. Embedded Computing: Arduino Tian, https://embeddedcomputing.weebly.com/arduino-tian.html, last accessed 2018/10/29.
16. Statista: Forecast on Connected Devices per Person Worldwide 2003-2020, https://www.statista.com/statistics/678739/forecast-on-connected-devices-per-person/, last accessed 2018/10/29.
17. Draishpits, G.: Peer-to-Peer Communication in Wireless Networks as an Alternative to Internet Access. Open University of Israel, Ra'anana (2016).
18. Suzuki, L.R.: Smart Cities IoT: Enablers and Technology Road Map. In: Rassia, S.T., Pardalos, P.M. (eds) Smart City Networks: Through the Internet of Things, Springer Optimization and Its Applications, vol. 125, pp. 167-190. Springer, Cham (2017).
19. Muhendra, R., Rinaldi, A., Budimana, M., Khairurrijal, K.: Development of WiFi Mesh Infrastructure for Internet of Things Applications. Procedia Engineering 170, 332-337 (2017).
20. Hrisko, J.: Using Raspberry Pi, HM-10, and Bluepy To Develop An iBeacon Mesh Network (Part 1), https://engineersportal.com/blog/2017/12/31/using-raspberry-pi-hm-10-and-bluepy-to-develop-an-ibeacon-mesh-network-part-1, last accessed 2018/10/29.
21. Commission on Science and Technology for Development, Nineteenth session: Smart cities and infrastructure, Report of the Secretary-General. United Nations, Geneva (2016).

22. Al-Hader, M., Rodzi, A.: The Smart City Infrastructure Development & Monitoring. Theoretical and Empirical Researches in Urban Management 4(2(11)), 87-94 (2009).
23. de Oliveira Neto, J.S., Kofuji, S.T.: Inclusive Smart City: An Exploratory Study. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2016, LNCS, vol. 9738, pp. 456-465. Springer, Cham (2016).
24. Bergenti, F., Poggi, A.: Multi-Agent Systems for the Application and Employing of E-Health Services. In: Cruz-Cunha, M.M., Tavates, A.J., Simoes, R. (eds.) Handbook of Research on Developments in E-Health and Telemedicine: Technological and Social Perspectives, pp. 247-264. IGI Global Publishing, Hershey, Pennsylvania (2010).
25. Chan, V., Ray, P., Parameswaran, N.: Mobile e-Health Monitoring: An Agent-Based Approach. IET Communication 2(2), 223-230 (2008).
26. ScalAgent Distributed Technologies: Benchmark of MQTT servers: ActiveMQ 5.10.0, Apollo 1.7, JoramMQ 1.1.3 (based on Joram 5.9.1), Mosquitto 1.3.5, RabbitMQ 3.4.2. ScalAgent Distributed Technologies, Échirolles (2015).
27. Mallick, S.: Histogram of Oriented Gradients, https://www.learnopencv.com/histogram-of-oriented-gradients/, last accessed 2018/10/29.
28. Salton, K.: Face Recognition: Understanding LBPH Algorithm, https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b, last accessed 2018/10/29.
29. Takuya, K., Akinori, H., Takio K.: Selection of Histogram of Oriented Gradients Features for Pedestrian Detection. In: Ishikawa M., Doya K., Miyamoto H., Yamakawa T. (eds.) ICONIP 2007, LNCS, vol. 4985, pp. 598-607. Springer, Berlin, Heidelberg (2008).
30. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2), 137-154 (2004).
31. Ilas, M.-E., Ilas, C.: A New Method of Histogram Computation for Efficient Implementation of the HOG Algorithm. Computers 7(1), 18 (2018).
32. Marco, V.R., Young, D.M., Turner, D.W.: The Euclidean Distance Classifier: An Alternative to the Linear Discriminant Function. Communications in Statistics – Computation and Simulation 16, 485-505 (1987).
33. Install OpenCV 3.2 Python/C++ on Raspberry PI, http://pklab.net/?id=392, last accessed 2018/10/29.
34. Rosebrock, A.: Raspbian Stretch: Install OpenCV 3 + Python on your Raspberry Pi, https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/, last accessed 2018/10/29.
35. Demaagd, K., Oliver, A., Oostendorp, N., Scott, K.: Practical Computer Vision with SimpleCV. O’Reilly Media, Sebastopol, CA, USA (2012).
36. Poggi, M., Mattoccia, S.: A Wearable Mobility Aid for the Visually Impaired based on Embedded 3D Vision and Deep Learning. In: 2016 IEEE Symposium on Computers and Communication Proceedings, pp. 208-213. University of Messina, Italy (2016).
37. Bai, J., Lian, S., Liu, Z., Wang, K., Liu, D.: Smart Guiding Glasses for Visually Impaired People in Indoor Environment. IEEE Transactions on Consumer Electronics 63(3), 258-266 (2017).
38. Zubov, D.: A Case Study on the Spatial Cognition of Surrounding Objects by the B&VI People Using Sound Patterns and Ultrasonic Sensing. In Kocovic, P., Behringer, R., Ramachandran, M., Mihajlovic, R. (eds.) Emerging Trends and Applications of the Internet of Things, pp. 105-116. IGI Global, Hershey, USA (2017).
39. Jacobson, R.D., Kitchin, R.M.: GIS and People with Visual Impairments or Blindness: Exploring the Potential for Education, Orientation, and Navigation. Transactions in GIS 2(4), 315-332 (1997).

40. Ramadhan, A.J.: Wearable Smart System for Visually Impaired People. Sensors 18(3), art. no. 843 (2018).
41. Gori, M., Cappagli, G., Tonelli, A., Baud-Bovy, G., Finocchietti, S.: Devices for Visually Impaired People: High Technological Devices with Low User Acceptance and no Adaptability for Children. Neuroscience & Biobehavioral Reviews 69, 79-88 (2016).
42. Wireless Communication between Raspberry Pi's via MQTT Broker/Client, https://tutorials-raspberrypi.com/raspberry-pi-mqtt-broker-client-wireless-communication/, last accessed 2018/10/29.
43. Fass, B.: A Heart Rate During a Heart Attack, https://www.livestrong.com/article/344010-the-normal-heart-rate-during-a-panic-attack/, last accessed 2018/10/29.
44. From KY-039 To Heart Rate, https://www.hackster.io/Johan_Ha/from-ky-039-to-heart-rate-0abfca, last accessed 2018/10/29.
45. Doukas, C.: Building Internet of Things with the Arduino. Amazon.com, CreateSpace Independent Publishing Platform, Seattle, Washington, USA (2012).
46. Kildall, S.: Raspberry Pi: Launch Python Script on Startup, https://www.instructables.com/id/Raspberry-Pi-Launch-Python-script-on-startup/, last accessed 2018/10/29.
47. Bennett, J.: Xamarin in Action: Creating native cross-platform mobile apps. Manning Publications Co., Shelter Island, NY, USA (2018).
48. HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi, https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi, last accessed 2018/10/29.
49. Nguyen, C.: Haptic Obstacle Detector for the Blind. KTH Royal Institute of Technology, Stockholm (2014).
50. Szeliski, R.: Computer Vision: Algorithms and Applications. 1st edn. Springer-Verlag, London (2011).
51. Rosebrock, A.: OpenCV Face Recognition, https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/, last accessed 2018/10/29.
52. Raspberry Pi Bluetooth In/Out Board Or "Who's Home", https://www.instructables.com/id/Raspberry-Pi-Bluetooth-InOut-Board-or-Whos-Hom/, last accessed 2018/10/29.
53. Lin, Y.W., Lin, C.Y.: An Interactive Real-Time Locating System Based on Bluetooth Low-Energy Beacon Network. Sensors 18(5), art. no. 1637 (2018).
54. Holton, B.: MyReader and MyEye from OrCam: Text and Item Recognition at the Touch of a Finger. AccessWorld 18(2), (2017), https://www.afb.org/afbpress/pubnew.asp?DocID=aw180205, last accessed 2018/10/29.
55. Campos, G.: Microsoft's Audio Wayfinding App for the Visually Impaired, https://www.avinteractive.com/news/ai/microsoft-launches-audio-wayfinding-app-visually-impaired-18-09-2018/, last accessed 2018/11/15.
56. Navigation Complex G2S, http://cputos.org.ua/technologies/navihatsijnyj-kompleks-g2s-go-to-the-sound-idy-na-zvuk/, last accessed 2018/11/15.
57. Peng, Y., Niu, X., Tang, J., Mao, D., Qian, C.: Fast Signals of Opportunity Fingerprint Database Maintenance with Autonomous Unmanned Ground Vehicle for Indoor Positioning. Sensors, 18(10), art. no. 3419 (2018).
58. Gajski, D.D., Abdi, S., Gerstlauer, A., Schirner, G.: Embedded System Design: Modeling, Synthesis and Verification. Springer-Verlag, New York (2009).
59. Warren, J., Gu, Y.: Pragmatic Health Information Technology Evaluation Framework. In: Handbook of eHealth Evaluation: An Evidence-based Approach, pp. 109-122. University of Victoria, Victoria, Canada (2016).