

A new system to detect coronavirus social distance violation

Marta Botella-Campos¹, José Luis Gacía-Navas², Albert Rego³, Sandra Sendra⁴, Jaime Lloret⁵

^{1,2,3,4,5}Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València, Valencia, Spain

⁴Department of Signal Theory, Telematics and Communications (TSTC), Universidad de Granada, Granada, Spain

Article Info

Article history:

Received Sep 9, 2020

Revised Jun 15, 2021

Accepted Jun 25, 2021

Keywords:

Face detection
Intelligent algorithm
Pandemic scenario
Physical distance
Smartphone
Sound detection

ABSTRACT

In this paper, a novel solution to avoid new infections is presented. Instead of tracing users' locations, the presence of individuals is detected by analysing the voices, and people's faces are detected by the camera. To do this, two different Android applications were implemented. The first one uses the camera to detect people's faces whenever the user answers or performs a phone call. Firebase Platform will be used to detect faces captured by the camera and determine its size and estimate their distance to the phone terminal. The second application uses voice biometrics to differentiate the users' voice from unknown speakers and creates a neural network model based on 5 samples of the user's voice. This feature will only be activated whenever the user is surfing the Internet or using other applications to prevent undesired contacts. Currently, the patient's tracking is performed by geolocation or by using Bluetooth connection. Although face detection and voice recognition are existing methods, this paper aims to use them and integrate both in a single device. Our application cannot violate privacy since it does not save the data used to carry out the detection and does not associate this data to people.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jaime Lloret
Instituto de Investigación para la Gestión Integrada de zonas Costeras
Universitat Politècnica de València
Valencia, Spain
Email: jlloret@dcom.upv.es

1. INTRODUCTION

Throughout history, different viral diseases happened to end up being considered global pandemics, causing miles of deaths. Among the most recent examples of the last 100 years, we can find the Spanish Flu in 1918 (H1N1), the swine flu in 2009 A(H1N1) [1] or recent coronaviruses such as the severe acute respiratory syndrome (SARS-CoV) in 2002 [2], Middle East respiratory syndrome-related coronavirus (MERS-CoV) in 2012 [3] or severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) in 2019 [4] with more than 1.9 million of infected people and near 120.000 deaths (up to April 2020).

The fast propagation of these viral diseases is due to its propagation being done by the air [5]. For this reason, the physical distance [6] between people and the use of medical masks are important actions to control the evolution of the pandemic. After overcoming the initial contagion outbreak of a pandemic, it is necessary to adopt these actions to avoid second outbreaks. However, in our daily routine, when we walk down the street and answer a phone call, we may not keep the recommended minimum physical distance. In this sense, we are more exposed to a possible contagion.

In our current society, we are surrounded by data networks, wireless sensor networks (WSNs) [7], and connected devices such as smartphones that greatly facilitate our daily tasks. This way of maintaining a

global network is known as internet of things (IoT) [8]. These types of networks are deployed in cities [9], endowing them with a certain intelligence and autonomy, in medical applications [10] or in our own homes to take care for our elderly people [11], among others.

The particular case of the development of applications and services for smartphones is generating great interest among researchers. One of the main factors that promote the development and improvement of these devices is related to the increase in the number of sales of mobile devices, although the sale of mobile devices begins to stabilize at approximately 1,500 million units as shown in Figure 1. Figure 1 shows the number of smartphones sold worldwide to end users from 2008 to 2018. As we can see, the growth in the number of sales has maintained an increasing trend, reaching 1,500 million units in 2018 [12]. Therefore, taking into account the statistics regarding the number of existing mobile devices, it makes sense to develop applications for smartphones that help us maintain the necessary physical distance to prevent possible contagions. The current computational power of these devices also allows running complex tasks such as image processing [13] or even audio equalization, as well as supporting many applications that were originally intended for desktop computers and could help us in our purpose.

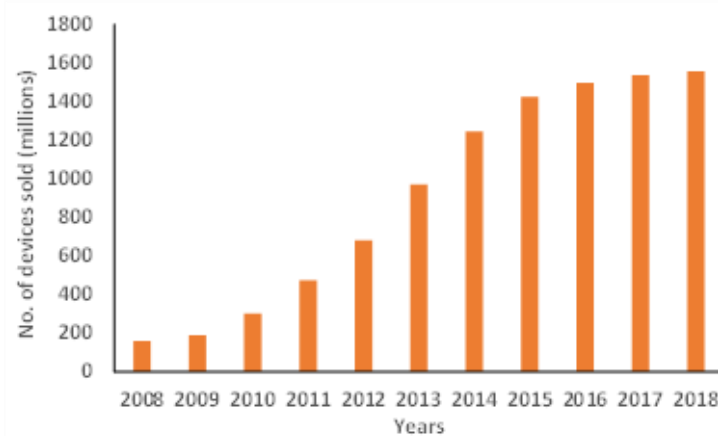


Figure 1. Effects of selecting different switching under dynamic condition

Taking into account the fact that almost everyone has at least one mobile device (especially in developed countries), we can think of developing an application for mobile terminals that allows us to define and keep the necessary social distance to prevent the rapid spread of the pandemic. Following this idea, we could use the sensors and actuators included in mobile phones to develop such systems [14]. After analysing the state of the art related to the development of mobile applications based on the use of sensors, we observed that beyond the development of bare-bones applications (in most cases, these applications do not use sensors) to know if a person is infected with COVID-19 or to follow their progress in the disease, no applications have been developed.

There are different ways for determining if other persons are nearby our position. It could be possible to use geolocation by GPS or 3G/4G for this purpose [15], [16]. However, this solution requires that the user whom we intend to locate has a mobile device and it would be necessary to enable the location of the terminal. We must also consider the precision error of these technologies which may be a few meters. We can conclude that these would not be optimal solutions. However, we can use 2 different procedures to determine if someone is too near to us. On the one hand, we can use our smartphone to determine if there is another voice source (other than ours) close to us. On the other hand, we could make use of the device's rear camera (for example, while we are making a call and we are not paying attention to our surroundings), which automatically analyses our nearby scenario and determines if a person is close by.

This paper proposes a new application to be run in a Smartphone to help us detect the presence of persons that are not maintaining the recommended physical distance. The proposed solution involves the development of 2 different applications to detect both the presence of a voice that does not correspond to the phone's owner and perform a face detection to verify the distance. The system was tested in a laboratory environment to check its feasibility and efficiency.

The rest of this paper is structured as follows; Section 2 presents some previous works related on the proposal where similar existing systems are presented. The proposed system is presented in section 3. In section 4 and 5, the implementation of the subsystems is explained. Section 4 describes the face detection

system. Section 5 shows the voice detection system. Other similar works are described in section 6 to compare their performance with our proposal. Finally, conclusion and future work are presented in section 7.

2. CAMERA AND MICROPHONE DETECTION RELATED WORKS

As Parra *et al.* explained [14], mobile phones are complex devices that contain several useful sensors to detect various parameters from our surroundings. This section presents some interesting works where existing methods and applications to perform face detection or voice detection are proposed. The presented applications will be based on the use of the camera and microphone embedded in smartphones.

2.1. Sound detector systems with mobile phones

Voice assistants such as Siri, Google Now, Cortana or Alexa, have turned smart devices and computers where touch interfaces cannot be used into very accessible systems [17], [18]. The use of intelligent algorithms complements its operations and can help us recognize the presence of people, performing authentication and so on. In 2017, Feng *et al.* [19] developed an authentication system called VAAuth able to provide continuous authentication for voice assistants. The system takes information of the body-surface vibrations of the owner and VAAuth continuously checks them to the received voice commands. VAAuth was implemented by using an accelerometer due to the low energy footprint and to collect on-body vibrations which the attacker cannot easily alter. In this way, VAAuth guarantees the voice assistant to execute only the commands that originate from the owner voice. VAAuth was tested with 18 users with several voice commands. Results show that VAAuth achieved 97% detection accuracy and less than 0.1% false positive rate. One of the main problems of VAAuth is the need of having 2 devices connected via Bluetooth. This fact implies that the overall energy consumption is high. Therefore, the voice capture depends on the availability of the second device which should be installed in glasses or should be worn as a gadget.

In this sense, it is recommended to have a unique device to perform the entire process such as the one presented by Bragg *et al.* [20]. In this case, authors presented a customizable mobile phone application able to detect sounds that hard-of hearing people need to know. To develop the system, users must train the application to identify the sounds users want to be able to detect by recording several examples of those sounds. The different sounds are then classified into categories of sounds. According to the authors, the system is flexible since it can include additional sounds by performing new training phases and, portable because the system is implemented in a mobile phone. Users are notified by using common mediums as text messages or vibrations. This application was checked through a Wizard-of-Oz in-lab and offline proof-of-concept for a Gaussian mixture model (GMM) based sound detection algorithm with different results for several sources and conditions. According to the system description, the smartphone should capture the audio and save it. Finally, the audio is analysed to determine which element generated that sound. It is not a real-time application that allows us to instantly detect and process audio.

Finally, it could be interesting to determine if the voice was generated by a person or the captured sound was a recorded voice-over. So, Zhang *et al.* [21] developed and tested the application VoiceLive. Authors evaluated a phoneme sound localization-based liveness detection application for mobile phones with the goal of differentiating between phrases spoken by a live user and a recorded one. The system was tested by replaying phrases by human speech production system and by mobile phone audio hardware. The application is based on the fact that modern mobile phones contain a stereo recording of dual microphones. The liveness detection application tries to detect the sound origin of each phoneme within the human vocal system and is based on the extraction of time difference of arrival (TDoA) of each phoneme sound to the two microphones of the phone. Authors tested the system in several persons and three different mobile phones as well as different sampling frequencies, phone displacements, and lengths of passphrases. Results showed that VoiceLive achieved over 99% detection accuracy and was able to determine if the captured sound came from a real person or not. However, the authors do not define whether it is possible to process the voice in real-time. Furthermore, it has not been proven that the system can correctly work when different voice sources are present.

2.2. Face detection and recognition with mobile phones

Face detection has aroused the interest of many researchers since it can be used to implement secure access systems based on mobile devices. Modern smartphones have RGB cameras suitable for capturing faces [22]. However, face biometric detection systems are not usually included as a native option. So, additional algorithms specifically designed for mobile environments are required [23].

In 2010, Dave *et al.* [24] analysed several algorithms to detect and recognize faces using mobile phones. Authors compared algorithms such as colour segmentation and template matching for face detection, and Eigenface and Fisherface for face recognition, to implement a face recognition system to be used on

DROID phones. From results, authors state that face detection algorithms usually present problems. For colour segmentation algorithm to work properly, the skin colour of a person should not be similar to the background colour. In this scenario, faces cannot be distinguished. Moreover, the template-matching algorithm is dependent on the standard template chosen and the illumination conditions, among others. Results showed that Eigenface and Fisherface achieved a total correct recognition/rejection rate of 84.3% with Eigenface, and 94.0% with Fisherface. Although the performance of the proposed system was quite high in terms of accuracy, the limitations of the face detection algorithm make it unsuitable for other scenarios where light conditions and background colours cannot be controlled. Furthermore, for the template matching algorithm to perform its task appropriately, the face needs to be centred and take a big part of the image. This situation implies that faraway faces will not be detected and thus, it will be impossible to determine at which distance an individual may be.

However, face detection algorithms have evolved over the past years. Elrefaei [25] presented a criminal detection framework to be used in Android mobile devices which aims to recognize the face of a criminal or a suspect. The application is based on the client-server paradigm that permits real-time face recognition in video surveillance. To develop the system, authors used the robust Viola-Jones algorithm since it is not affected by illumination conditions. The application also includes an extension for face tracking based on an optical flow algorithm with fast corner features and regular features, as extraction methods. The application was tested using Sony Xperia Z2 Android 5.1 Lollipop Smartphone and results showed that the presented system achieves a higher level of accuracy and efficiency than optical flow with fast corner features. In spite of the fact that this proposal achieves acceptable performance rates, this solution requires a server side to perform its task. Moreover, face detection was performed on real-time captured videos, which increases the computational power. Thus, this application is still not appropriate for the aim of this project.

Finally, Yang and Luo [26] proposed a real-time face recognition system on Android platform able to perform face detections by applying AdaBoost algorithm and face recognition by utilizing Eigenfaces. Authors also applied some methods to speed up the processes of face detection and recognition and enhance the rate of successful recognitions. From results, authors concluded that their system could perform real-time face recognition/detection over Android Operative System and mobile devices in over 300 ms. Though this system manages to perform real-time face detection and recognition in a short time, the proposed solution was designed as an identification tool, and requires a database in which the user can store face samples. Again, we encounter that the proposed solution operates correctly when being fed with multiple images to build a strong classifier of face features.

Except for very few works like the one presented by Kim [27], the existing systems only base their use on one type of detection. Our proposal combines the use of sound and image sources to verify and estimate people's presence and distance to the user's terminal to ensure physical distance and minimize contagion by viruses transmitted by air. The proposed application improves the presented works in different aspects. On the one hand, we present an application that bases its operation on the use of embedded sensors in the smartphone, i.e., the microphone and the camera. Thus, avoiding the dependency on other devices to capture video/audio. The application can recognize the owner's voice by using the mel frequency cepstral coefficients (MFCC) [28]. Consequently, the presence of other voices will imply a possible violation of social distance. In addition, voice and image processing is performed in real-time, and therefore notifications to the users will be more effective. To correctly perform both voice and image processing, the application uses deep learning techniques [29] that will help us have a more accurate detection. Finally, the proposed system ensures users' privacy since it does not collect any information from the terminal and does not need Internet connection nor GPS coordinates. Therefore, the use of this approach will always be local and anonymous.

3. PROPOSAL DESCRIPTION AND METHOD

This work has been done with the following method. Firstly, the research problem indicated in the previous sections was identified. Then, the hypothesis of creating a system based only on the microphone and the camera that can detect social distance violations was specified. In this section, the system is explained in terms of architecture, algorithm and working procedure.

Figure 2(a) shows the different modules of the software architecture. The main algorithm, which is explained in this subsection, uses two subsystems to work. First, the voice detection subsystem analyses the sound captured by the microphone to detect unknown voices. Secondly, the face detection system is the subsystem that uses the camera to detect how many faces are close to the user. The system reads the number of faces and if there are unknown voices using the microphone. The main system must provide the voice detection subsystem with samples to train the algorithm.

The system then uses both the microphone and the camera to detect possible violations of the security distance. However, both systems are not used at the same time, meaning that there is no necessity to synchronize both systems. Figure 2(b) shows the state diagram of the alarm detection system on which transition from recognizing voice to detecting faces and vice versa are explained. States' names are displayed in black while transitions are blue. The transitions arcs show the condition and the action performed. Firstly, when the mobile is in idle state or the user is using it for surfing on the Internet or using other apps, only the microphone is activated. In the figure, it would be the "analysing microphone data" state. In this state, the data of the microphone is being analysed to detect voices. The voice of the user will be detected but if an unknown voice (the voice of a second person) is detected, an alarm will be triggered. The system is now in "risk state" which means that the security distance is not being implemented. When the risk state finishes, the system returns to the previous state, in this case, "analysing microphone data".

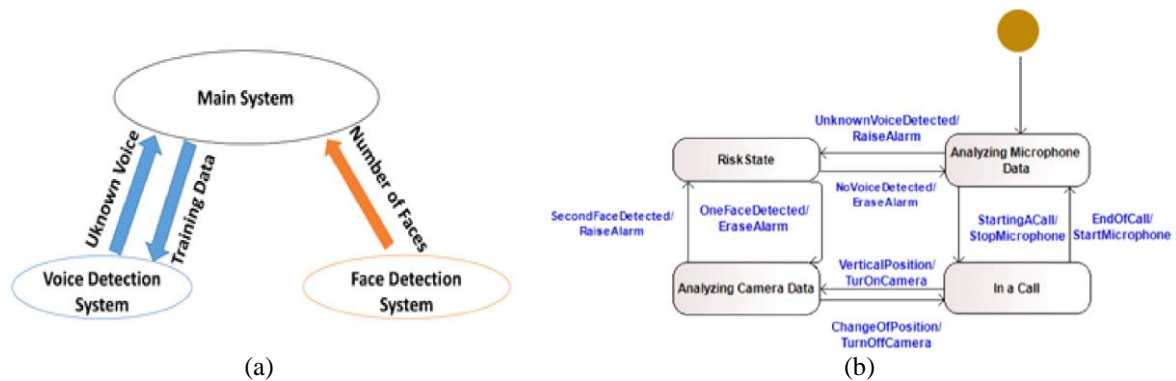


Figure 2. Architecture proposal, (a) software architecture and information exchange of the different software modules of the system, (b) state machine diagram of the main algorithm

The other possibility to detect persons is using the camera. This is used only during a call. Therefore, when a call is performed (incoming or outgoing), the system moves to a new state "in a call" where the microphone is not used anymore, and the phone is waiting to use the camera. However, the camera will only be analysed when the phone is detected to be in a vertical position. When the accelerometer detects a vertical position, the system state will change to "analysing camera data". Now the system is searching for faces in the camera. Here the number of faces is the important value. If there are two different faces detected by the camera, the alarm would be activated. In this case, the system would enter the "risk state" state. The difference is that when the alarm ends, the new state will be "analysing camera data" instead of "analysing microphone data". Then, the current state can move backwards to "in a call" or even to "analysing microphone data" if the cell phone is detected as in a non-vertical position or in a "not-in-a-call" state. This is an infinite loop that is repeated while the user does not close the app.

The algorithm of the main system is shown in algorithm 1. Firstly, the variables are initialized. In order to represent the states, a numeric value is assigned to them. The state "analysing microphone data" is number 0, "in a call" is number 1, "analysing camera data" is number 2 and, finally, "risk state" is number 3.

The first time the application is run, the user will be required to record some voice samples to start the voice recognition training process. In this process, the user must say several sentences (the minimum number of sentences is 5) to train the voice detection subsystem. However, there is no need for a face detection training process since the application will never capture the user's face. The face detection process only works when the user is talking on the phone. Therefore, the camera only captures images of the surrounding area but not the user's face. This is the reason why the app does not need to differentiate between the owner of the phone and other people.

Once the training is complete, the application enters in an infinite loop until the user exits or closes the application. Starting in the state 0, the conditions shown in Figure 2(b) are evaluated to update the "currentStatus" variable, which indicates the current status. Depending on the phone status and the values that the subsystems return, the current state changes. When the "risk state" is reached, the alarm is activated. After that, when the alarm ends, (it is discarded by the user or a certain amount of time has passed), the system returns to its previous state. This is done using the "riskStateEntry" variable, which stores the last state before moving into the "risk state". Between iterations, the app will be paused during a certain time (by default, 200 ms).

Algorithm 1. Main algorithm

```

userExit = 0
currentStatus = AnalMicroData
riskStateEntry = 0
UserVoice = StartTrainingProcess()
Do
  If(currentStatus = 0)
    If(GetStatusMicrophone()= Disabled)
      StartMicrophone()
      StartVoiceRecognition(UserVoice)
    Else
      If (ReadVoiceRecognition()= True)
        currentStatus = 3
        riskStateEntry = 0
        RaiseAlarm()
      Else
        If(GetPhoneStatus()= InCall)
          currentStatus = 1
          StopMicrophone()
        End If
      End If
    End If
  End If

  If (currentStatus = 1)
    If(GetPhoneStatus() != InCall)
      currentStatus = 0
    Else
      If(ReadPosition()= VERTICAL)
        currentStatus = 2
        riskStateEntry = 0
        RaiseAlarm()
      End If
    End If
  End If

  If (currentStatus = 2)
    If(ReadPosition() != VERTICAL))
      currentStatus = 1
      TurOffCamera()
    Else
      If(GetStatusCamera()= Disabled)
        TurOnCamera()
      End If
      If (ReadNumberFaces() > 1)
        currentStatus = 3
        riskStateEntry = 2
        RaiseAlarm()
      End If
    End If
  End If

  If (currentStatus = 3)
    If(ReadAlarmState() != Raised)
      currentStatus = riskStateEntry
    End If
  End If
  Sleep(deltaTime)
  userExit = ReadUserExit()
While userExit = 0

```

4. FACE DETECTION

In this section, the part of the application that is able to detect people's faces while the owner of the phone is talking on the mobile phone, is explained. The main two parts of this application are the accelerometer and the face detection part. When there is an incoming call, the app will first detect the position of the phone by using the accelerometer, to know whether the person is holding the phone while talking or if the phone is leaning on the table or any other horizontal surface. In case the phone is not in horizontal position, the camera will be activated. Taking into account that the mobile phone is not in horizontal position and there is an incoming call, there are two possible options on how the person is talking on the phone: using headphones or holding the phone by hand. If the camera detects that the light intensity

received is very low (dark image) the app will suppose that the user is using headphones and no face detection will be activated. In any other case face detection will be activated.

4.1. Accelerometer

Every mobile phone has acceleration sensors known as accelerometers [30]. This sensor measures in m/s^2 the acceleration force applied to a device in the three physical axes (x, y, z), including the force of gravity. Algorithm 2 explains the access to the accelerometer sensor of Android mobiles and how to use it in order to obtain the position of the phone. Algorithm 2 is the part of the code that is included in the onCreate function on MainActivity, that is the first function when running the app. In this code, the SensorManager system service is accessed first. This system allows access to the accelerometer information that we have saved on the "sensor" variable. In case the "sensor" variable is correctly declared and is not null, SensorEventListener and onSensorChanged functions are declared. This function is accessed every time the accelerometer sensor detects any change. The x, y, z values can be taken as shown inside of onSensorChanged function. In case x and y values are between -3 and 3, the mobile phone will be horizontal. In any other case, the phone position will be supposed.

Algorithm 2. Access to accelerometer sensor and mobile position detection by using x, y, z acceleration

```

sensorManager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
sensor = sensorManager!!.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
if (sensor == null) finish()
sensorEventListener = object : SensorEventListener {
    override fun onSensorChanged(sensorEvent: SensorEvent) {
        val x = sensorEvent.values[0]
        val y = sensorEvent.values[1]
        val z = sensorEvent.values[2]
        If (x < 3 && x > -3 && y < 3 && y > -3)
            posicion.text = "Horizontal"
        Else
            posicion.text = "Talking on phone"
        End If
    }
    override fun onAccuracyChanged(sensor: Sensor, i: Int) {}
}

```

4.2. Face detection

Once the position of the phone is defined the camera will be activated. First of all, light intensity will be captured in order to know if the camera can be used to detect faces or not. If the light intensity is high enough, we will suppose that the person is holding the phone and the face detection process will start. Figure 3 shows the process, explained before, to decide whether to activate or not the face detection process.

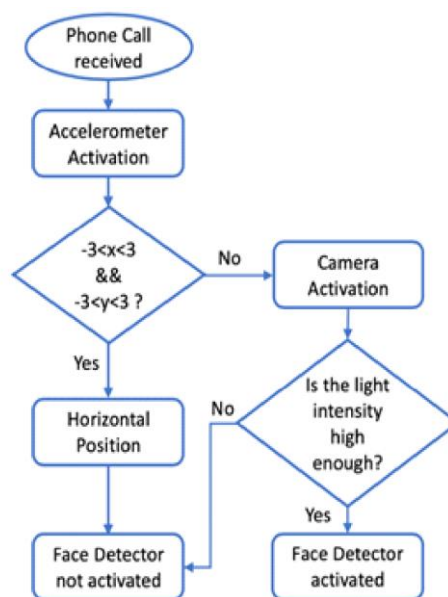


Figure 3. Face detector activation flowchart

In order to develop the real time face detection app, we used a GitHub repository [31] that uses the Firebase Platform Services to develop applications [32]. Thanks to the services of Firebase Platform, it is possible to develop an Android Mobile application to detect facial features. The application detects every face captured by the camera and draws a square around them. In order to draw these squares, the application measures the size of the faces by detecting different facial features of all people captured by the camera. Depending on the shape of people’s heads, the measurement of the square’s sides can vary for the same measurement distance.

We can obtain the size of the side of every square surrounding a head. The size of these squares can be obtained as an internal Android unit that is used to draw the squares around the faces. We used the side measurement of the squares to establish whether people around the mobile owner are far enough according to the security distance established for COVID-19 disease [33]. Figure 4 shows the data obtained after testing the application with four different persons in order to know how measurements vary depending on the distance between the phone and the person that is being captured.

The data obtained shows that the variation of the side measurement is higher for short distances than for large distances. Figure 4 also shows that the application works correctly with different people. Data shown in Figure 4 has been obtained by capturing people that are directly looking at the phone camera, though in real situations people will not be directly looking at the phone camera. In order to know how the orientation of people’s faces can affect to their face size measurement, we have measured the same four persons looking to -90° (left side face), -45° , 0° (frontal position), 45° and 90° (right side face) degrees from the phone at 1 meter and 1.5 meters from the phone.

Table 1 shows the average error of the square side measurement when people are looking -90° , -45° , 45° and 90° degrees from the phone, compared to frontal position. Taking into account that the square side measurement variations between every 25 cm are: 75-100 cm=-68.08 units; 100-125 cm=-36.92 units; 125-150 cm=-30.00 units; 150-175 cm=-20.42 units; and keeping in mind the average error shown in Table 1, we can say that it is possible to detect if people are far enough or not.

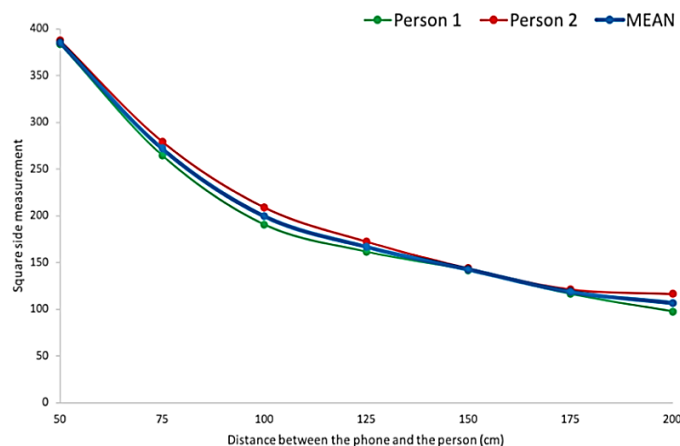


Figure 4. Square side measurement depending on the distance

Table 1. Average error of square side measurement

Average Error at 1 meter from the phone		Average Error at 1.5 meters from the phone	
Person Position	Error	Person Position	Error
-90	17,66	-90	10,50
-45	7,17	-45	4,33
45	4,42	45	2,08
90	17,08	90	8,58

5. OWNER VOICE RECOGNITION

Speaker recognition is a challenging task that aims to extract speech characteristics to identify a user. Typically, a speaker recognition system consists of two modules: Feature extraction and feature matching [34]. This way, small amounts of data that represent each speaker will be extracted from the voice signal to compare those features with new inputs. This section presents the process used to identify a speaker within an Android application. Thus, it is divided into three different subsections.

5.1. Application structure on android

For the aim of this project, we built an android application able to extract voice features of a user and identify his/her voice using machine learning algorithms. Figure 5 shows the flow diagram of this application. As the figure shows, once the application is started, the user will be asked to record 5 samples of their voice. At each sample, mel frequency cepstral coefficients [35] will be extracted along with the central frequencies of the mel filter banks. This data will then be exported into a csv file to allow further processing before training the machine learning model. Once the model has been trained, the user will be able to record a new voice sample that will be compared to the previous ones and classified in two different groups: Known user or unknown speaker.

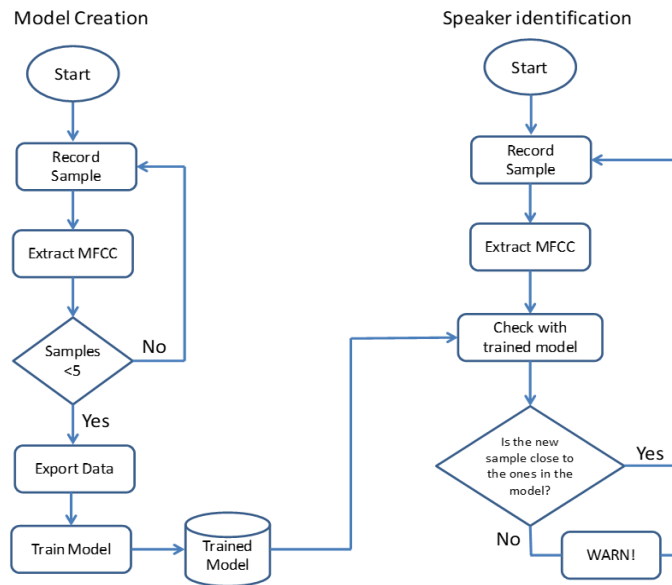


Figure 5. Application flowchart

5.2. Mel frequency cepstral coefficients

One of the most important features used in speech and speaker recognition systems are MFCCs [36]. This technique is based on human hearing behaviour and thus, it expresses frequencies in the MEL scale. Figure 6 shows the overall process of extracting cepstral coefficients. In this project, we used the TarsosDSP library to perform the analysis of the data on Android systems [37]. This library allows real-time feature extraction and synthesis both in audio records and videos. Following the indications in [38], cepstral coefficients were extracted configuring TarsosDSP method to extract 13 coefficients using 28 filters in the frequency range from 300 Hz to 8000 Hz, for a sampling rate of 44.1 kHz and a buffer size of 2048 samples.

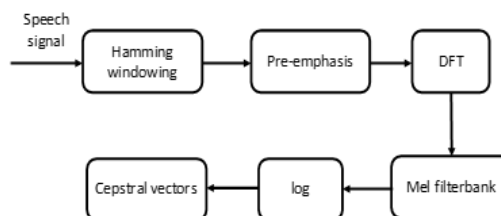


Figure 6. MFCC block diagram

5.3. Deep learning algorithm

In order to classify new voice samples, we created a deep learning architecture using Deeplearning4j library [39]. This way, we created a multi-layered artificial neural network. The proposed prediction model structure is composed of 3 layers. The Input layer was configured to have 13 input nodes that will pass the

collected data to a Hidden layer with 6 input nodes and 3 output nodes. Finally, the Output layer is formed of a single node to classify data into two categories: Known user or unknown speaker. In our case, we used the negative log-likelihood (NLL) cost function to optimize the parameter values given to the neural network model. Moreover, we implemented backpropagation in order to minimize the error on the computation of the gradient performed by previous layers. The training of the model was performed by recording five samples of the user's voice that will be passed 4 times to achieve high accuracy results.

5.4. Owner identification

Figure 7 shows the utterance of the sentence "Hello, my name is Josefina" performed by two different speakers. As the figure shows, the voiceprint of each person is unique. When comparing voice biometrics, it is possible to detect and identify different speakers. By storing multiple voiceprints of the end user during the enrollment phase, we created a model that will be used to verify the presence of other individuals that may not maintain the recommended physical distance. The match score given by the classification algorithm returns an estimation value, resulting in a near 0 value when the speaker is the owner of the terminal, and a near 1 value when the speaker is unknown to the system. Thus, extracting features of a user's voice can be of great use in authentication methods where fingerprint or facial recognition are impossible, like customer-support calls.

Speaker verification can be clearly seen in Figure 8, where cepstral coefficients of the above-mentioned voiceprints have been plotted. The Mel frequency cepstrum is often used in sound processing to represent the power spectrum of a sound in an equally spaced mel scale to approximate it to the human auditory system's response. These coefficients are used as features in speech recognition to help lessen the influence of noise and extract relevant audio characteristics that are necessary for the speech recognition process. When comparing the performances of both speakers, one can observe that the rate, the periodicity, and complexity of vibration of each speaker is different. Thus, extracting the cepstral coefficients of the end user allows us to identify a person among the rest.

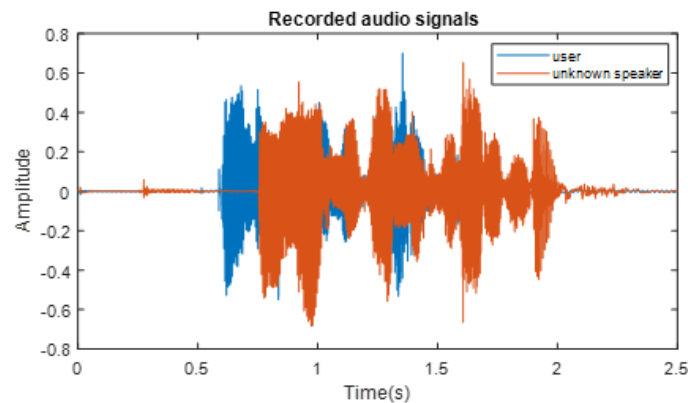


Figure 7. Recorded samples of two speakers saying the same sentence

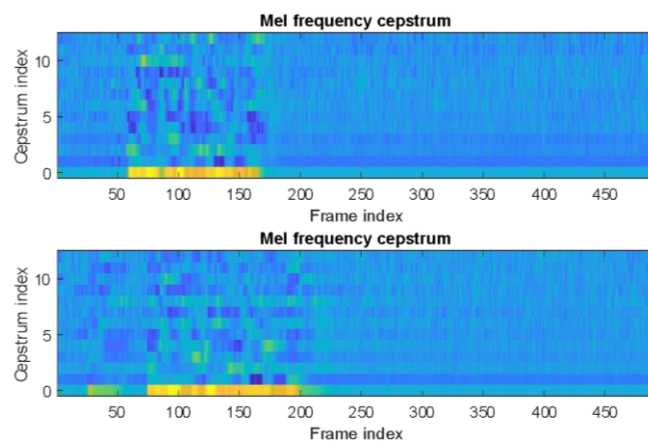


Figure 8. Cepstral coefficients of the recorded samples of two different speakers

Table 2 shows the mean estimation values of the classification algorithm after recording 5 samples for a known user and comparing new samples with the built model. These values were obtained by performing the mean value of three attempts for every speaker. As the results show, by extracting the cepstral coefficients of a speaker to train a model we can successfully differentiate among different speakers to determine whether a person is accompanied by someone else or, in this case, whether other individuals may be too close to the user's terminal.

Table 2. Estimation values of the classification algorithm

Speaker	Mean Estimation Value
Known user	0.26
Unknown speaker 1	0.85
Unknown speaker 2	0.76
Unknown speaker 3	0.82

6. OTHER SIMILAR SOLUTIONS

Once the design and results of our proposal have been discussed, the system will be compared with other implemented solutions. In this section, a brief comparison is presented and the differences between our proposal and other alternatives are summarized. Although the COVID-19 it is a relevant topic, there are not numerous technical solutions regarding social distance. Table 3 shows a comparison between the proposal and the alternatives.

Table 3. Comparison of alternatives

Related solutions	Scope	Sensors and data used	Needs other users using the same system?	Identification	Privacy
[37]	Individual	Bluetooth Device-to-device interaction	Yes	Needs a user profile of your health system	Uses sensitive data
[38]	Individual	Bluetooth	Yes	Needs your actual data	Uses sensitive data
[39]	Individual	Bluetooth	No	None	No data required
[40]	Global	GPS	-	None	Uses anonymous location and mobility data
[41]	Global	Location Mobility GPS	-	None	Uses anonymous location and mobility data
[42]	Place	Location Mobility GPS	Yes	User account	Uses location and mobility data
[43]	Individual	User Feedback Camera	No	Snapchat user account	No data required
This work	Individual	Feet detection Speech recognition Camera Facial detection	No	None	No data required

Apple and Google have launched an app to monitor people's contacts using Bluetooth [40]. The app tracks data about the user and the location, raising some questions about privacy, especially taking into account that they operate with sensitive data, being able to categorize if a user is infected. The Australian government proposed COVIDSafe [41], an application that uses Bluetooth to detect other users who have been in contact with COVID. The contacts are used to trace a possible spread when the user is infected. However, it does not alert the user from a social distance violation. Another solution was developed by the United Nations Technology Innovation Labs (UNTIL) [42]. Their app also uses Bluetooth in a similar way. However, it detects every type of device with Bluetooth, so it can warn of a device-to-device distance violation, but not of a social distance violation.

In a more global way, services like Unacast's Social Distancing maps [43] or Inoco [44] work with the location data of a wide number of users to show a statistical analysis of the mobility and the social distance violation rate in different countries. They offer a software development kit (SDK) that third parties can include in their applications. Therefore, they collect anonymous data and process it.

Some solutions such as Crowdless, developed by Lanterne [45], are used to alert people of crowded areas. The user can check the state of occupation of the place they want to go to before actually doing so. The application uses location data and users' feedback to determine how crowded a place is. Finally, the only

known solution to who uses the camera is a filter of "Snapchat" social network application that adds a visual warning when it detects a person less than two meters away [46]. Though it is only a complement of the application, and not a specialized application, and a Snapchat account is required.

In comparison, our proposal provides a solution that respects the privacy of the user and their anonymity, not using identification, sensitive data nor locations as parameters. Moreover, it combines the use of the camera with the use of speech recognition techniques. Consequently, it is able to warn the user of social distance violations without storing any private data. Furthermore, it warns the user without needing to interact with other devices or users of the app, notifying the social distance violations only to the end user.

The natural evolution of this kind of applications and systems is to develop a common platform able to obtain data from different patient's profiles to analyse them [47]. Because the environments where the measures are taken can be quite different, it is also important to think of a versatile system able to be adapted to the features of each scenario [48]. Finally, a more accurate mechanism to recognize the owner and the rest of people can be necessary [49], [50].

7. CONCLUSION AND FUTURE WORK

In this work, a system to detect people that may be around the user is presented. The proposal is based on two different detection systems combined and synchronized. On the one hand, the face detection system, used to detect people using the phone camera during a call. The performed tests proved that the system can be used to detect nearby people. On the other hand, the voice recognition system. This module is based on a deep neural network, and uses NLL for optimization. The system is trained when the application is run for the first time. The results showed that the system obtains different coefficients for different speakers that can be used to identify unknown speakers. However, due to restrictions related to the pandemic, more work needs to be carried out in real time scenarios and crowded environments.

In future works, the system will be improved. First, the use of geolocation-related data will be studied to gather statistics, perform studies, and generate maps of the confirmed cases at a city level. Moreover, the data can be used to raise alarms to the closest systems and entities. Another possible improvement is regarding the intelligent system for voice recognition. In the future, the system could use an online algorithm to enhance the functionality and improve its accuracy. Moreover, it could be trained prior to its installation to achieve better results. Another improvement would be adapting the system to other environments and issues, such as noisy environments, by enhancing the subsystems and considering different constraints and casuistics. As for face detection, curvelet transform and support vector machine (SVM) classifiers could be used as a mean to extract face features, and detect whether the same person keeps approaching a user. Finally, the app will be prepared to be uploaded to a platform such as GitHub for further improvements through user comments and testing.

ACKNOWLEDGEMENTS

This work has been partially supported by the "Ministerio de Educación, Cultura y Deporte", through the "Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2015)", grant number FPU15/06837, and by "Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2019)", grant number FPU19/04023.

REFERENCES

- [1] World Health Organization (WHO), "Influenza: are we ready?," Accessed on April 11, 2020. [Online]. Available: <https://www.who.int/influenza/spotlight>
- [2] World Health Organization (WHO), "Severe Acute Respiratory Syndrome (SARS)," Accessed on April 11, 2020. [Online]. Available: <https://www.who.int/ith/diseases/sars/en/>
- [3] World Health Organization (WHO), "Middle East respiratory syndrome coronavirus (MERS-CoV)," accessed on April 11, 2020. [Online]. Available: [https://www.who.int/news-room/fact-sheets/detail/middle-east-respiratory-syndrome-coronavirus-\(mers-cov\)](https://www.who.int/news-room/fact-sheets/detail/middle-east-respiratory-syndrome-coronavirus-(mers-cov)),
- [4] World Health Organization (WHO), "Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)," Accessed on April 12, 2020. [Online]. Available: [https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it)
- [5] L. Li *et al.*, "Propagation analysis and prediction of the COVID-19," *Infectious Disease Modelling*, vol. 5, pp. 282–292, 2020, doi: 10.1016/j.idm.2020.03.002.
- [6] World Health Organization (WHO), "Coronavirus disease 2019 (COVID-19) Situation Report –66," Accessed on April 12, 2020. [Online]. Available: https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200326-sitrep-66-covid-19.pdf?sfvrsn=81b94e61_2

- [7] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *Applied System Innovation*, vol. 3, no. 1, 2020, doi: 10.3390/asi3010014.
- [8] A. Paul and R. Jeyaraj, "Internet of Things: A primer," *Human Behavior and Emerging Technologies*, vol. 1, no. 1, pp. 37-47, 2019, doi: 10.1002/hbe2.133.
- [9] F. Luna, J. F. Valenzuela-Valdes, S. Sendra, and P. Padilla, "Intelligent wireless sensor network deployment for smart communities," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 176-182, 2018, doi: 10.1109/MCOM.2018.1700553.
- [10] S. Sendra, L. Parra, J. Lloret, and J. Tomás, "Smart system for children's chronic illness monitoring," *Information Fusion*, vol. 40, pp. 76-86, 2018, doi: 10.1016/j.inffus.2017.06.002.
- [11] J.J. Rodrigues *et al.*, "Enabling technologies for the internet of health things," *IEEE Access*, vol. 6, pp. 13129-13141, 2018, doi: 10.1109/ACCESS.2017.2789329.
- [12] Statista, "Number of smartphones sold to the end user globally from 2011 to 2018," Accessed on April 15, 2020. [Online]. Available: <https://es.statista.com/estadisticas/521667/numero-de-smartphones-vendidos-en-el-mundo-al-usuario-final/#statisticContainer>,
- [13] C. M. Ranieri, S. MacLeod, M. Dragone, P. A. Vargas, and R. A. F. Romero, "Activity Recognition for Ambient Assisted Living with Videos, Inertial Units and Ambient Sensors," *Sensors*, vol. 21, no. 3, 2021, doi: 10.3390/s21030768.
- [14] L. Parra, S. Sendra, J. M. Jiménez, and J. Lloret, "Multimedia sensors embedded in smartphones for ambient assisted living and e-health," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13271-13297, 2016, doi: 10.1007/s11042-015-2745-8.
- [15] Y. Liu, Z. Ai, G. Liu, and Y. Jia, "An integrated shark-fin antenna for MIMO-LTE, FM, and GPS applications," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 8, pp. 1666-1670, 2019, doi: 10.1109/LAWP.2019.2927019.
- [16] S. Chen, B. Shen, X. Wang, and S. J. Yoo, "Geo-location information aided spectrum sensing in cellular cognitive radio networks," *Sensors*, vol. 20, no. 1, 2020, doi: 10.3390/s20010213.
- [17] M. B. Hoy, "Alexa, Siri, Cortana, and more: an introduction to voice assistants," *Medical reference services quarterly*, vol. 37, no. 1, pp. 81-88, 2018, doi: 10.1080/02763869.2018.1404391.
- [18] A. L. Nobles, E. C. Leas, T. L. Caputi, S. H. Zhu, S. A. Strathdee, and J. W. Ayers, "Responses to addiction help-seeking from Alexa, Siri, Google Assistant, Cortana, and Bixby intelligent virtual assistants," *NPJ digital medicine*, vol. 3, no. 1, pp. 1-3, 2020, doi: 10.1038/s41746-019-0215-9.
- [19] H. Feng, K. Fawaz, and K. G. Shin, "Continuous authentication for voice assistants," *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCOM'17)*, Snowbird, Utah (USA), 2017, pp. 343-355, doi: 10.1145/3117811.3117823.
- [20] D. Bragg, N. Huynh, and R. E. Ladner, "A personalizable mobile sound detector app design for deaf and hard-of-hearing users," *Proceedings of the 18th International Conference on Computers and Accessibility (ASSETS'16)*, Reno, Nevada (USA), 2016, pp. 3-13, doi: 10.1145/2982142.2982171.
- [21] L. Zhang, S. Tan, J. Yang, and Y. Chen, "Voicelive: A phoneme localization-based liveness detection for voice authentication on smartphones," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, Vienna (Austria), 2016, pp. 1080-1091, doi: 10.1145/2976749.2978296.
- [22] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, 2020, Art. no. 1188, doi: 10.3390/electronics9081188.
- [23] A. Rattani and R. Derakhshani, "A survey of mobile face biometrics," *Computers and Electrical Engineering*, vol. 72, pp. 39-52, 2018, doi: 10.1016/j.compeleceng.2018.09.005.
- [24] G. Dave, X. Chao, and K. Sriadibhatla, "Face recognition in mobile phones," Department of Electrical Engineering Stanford University, Stanford (USA), pp. 7-23, 2010. [Online]. Available: https://stacks.stanford.edu/file/druid:rz261ds9725/Sriadibhatla_Davo_Chao_FaceRecognition.pdf
- [25] L. A. Elrefaei, A. Alharthi, H. Alamoudi, S. Almutairi, and F. Al-rammah, "Real-time face detection and tracking on mobile phones for criminal detection," *Proceedings of the 2nd International Conference on Anti-Cyber Crimes (ICACC'17)*, Abha (Saudi Arabia), 2017, pp. 75-80, doi: 10.1109/Anti-Cybercrime.2017.7905267.
- [26] G. Yang and J. N. Luo, "A real-time face recognition system for android smart phone," *Advanced Materials Research*, vol. 756, pp. 4006-4010, 2013, doi: 10.4028/www.scientific.net/AMR.756-759.4006.
- [27] T. Kim, H. Park, S. H. Hong, and Y. Chung, "Integrated system of face recognition and sound localization for a smart door phone," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 3, pp. 598-603, 2013, doi: 10.1109/TCE.2013.6626244.
- [28] L. Muda, M. Begam and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *Journal of Computing*, vol. 2, no. 3, pp. 138-143, 2010.
- [29] T. B. Mokgonyane, T. J. Sefara, T. I. Modipa, M. M. Mogale, M. J. Manamela, and P. J. Manamela, "Automatic speaker recognition system based on machine learning algorithms," in *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 2019, pp. 141-146, doi: 10.1109/RoboMech.2019.8704837.
- [30] A. R. Javed, M. U. Sarwar, S. Khan, C. Iwendi, M. Mittal, and N. Kumar, "Analyzing the effectiveness and contribution of each axis of tri-axial accelerometer sensor for accurate activity recognition," *Sensors*, vol. 20, no. 8, p. 2216, 2020, doi: 10.3390/s20082216.
- [31] A real-time face detection Android library, "Android Face Detector," Accessed April 17, 2020. [Online]. Available: <https://github.com/husaynhakeem/android-face-detector>.

- [32] S. Milovanović, Z. Bogdanović, T. Naumović, D. Barać, and M. Despotović-Zrakić, "Analysis of Firebase platform in mobile application development," *Info M*, vol. 17, no. 65, pp. 17–25, 2018.
- [33] C. Sun and Z. Zhai, "The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission," *Sustainable Cities and Society*, vol. 62, 2020, doi: 10.1016/j.scs.2020.102390.
- [34] A. Mohamed, "Deep Neural Network Acoustic Models for ASR," Doctor of Philosophy Graduate Department of Computer Science, University of Toronto, 2014.
- [35] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques," *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, Cholula, Puebla, 2012, pp. 248–251, doi: 10.1109/CONIELECOMP.2012.6189918.
- [36] Q. Mei, M. Güll, and M. Boay, "Indirect health monitoring of bridges using Mel-frequency cepstral coefficients and principal component analysis," *Mechanical Systems and Signal Processing*, vol. 119, pp. 523–546, 2019, doi: 10.1016/j.ymssp.2018.10.006.
- [37] Github, "JorenSix/TarsosDSP," 2015. Accessed April 17, 2020. [Online]. Available: <https://github.com/JorenSix/TarsosDSP>
- [38] V. Tiwari, "MFCC and its applications in speaker recognition," *International Journal on Emerging Technologies*, vol. 1, no. 1, pp. 19–22, 2010.
- [39] Deeplearning4j.org, "Deeplearning4j," 2020. Accessed April 17, 2020. [Online]. Available: <https://deeplearning4j.org>
- [40] "Apple and Google are building a coronavirus tracking system into iOS and Android." Accessed May 11, 2020. [Online]. Available: <https://www.theverge.com/2020/4/10/21216484/google-apple-coronavirus-contract-tracing-bluetooth-location-tracking-data-app>
- [41] Department of Health, Australian Government "COVIDSafe app." Accessed July 8, 2020. [Online]. Available: <https://www.health.gov.au/resources/apps-and-tools/covidsafe-app>
- [42] "This app can help you perform social distancing." Accessed May 11, 2020. [Online]. Available: <https://indianexpress.com/article/technology/tech-news-technology/how-to-maintain-social-distancing-app-download-6384726/>
- [43] "COVID-19 Location Data Toolkit," Unacast. Accessed May 11, 2020. [Online]. Available: <https://www.unacast.com/covid19>
- [44] "Location data to control COVID-19 while respecting individual privacy." Accessed May 11, 2020. [Online]. Available: <https://www.inloco.com.br/en/>
- [45] "Social distancing app uses space to save lives." Accessed May 11, 2020. [Online]. Available: https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Social_distancing_app_uses_space_to_save_lives
- [46] O. Oakes, "We Are Social deploys Snap AR lens to help people with social distancing" 2020. Accessed May 11, 2020. [Online]. Available: <https://www.campaignlive.co.uk/article/social-deploys-snap-ar-lens-help-people-social-distancing/1679294>
- [47] D. C. Tran, "An open toolbox for generating map of actively confirmed SARS-CoV-2 or COVID-19 cases in Vietnam," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 6, pp. 2396–2403, 2020, doi: 10.11591/eei.v9i6.2621.
- [48] A. O. A. Noor, "Robust speaker verification in band-localized noise conditions," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 13, no. 2, pp. 499–506, 2019, doi:10.11591/ijeecs.v13.i2.pp499-506.
- [49] H. Zhi and S. Liu, "Face recognition based on genetic algorithm," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 495–502, 2019, doi: 10.1016/j.jvcir.2018.12.012.
- [50] M. T. Ghazal and K. Abdullah, "Face recognition based on curvelets, invariant moments features and SVM," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 2, pp. 733–739, 2020, doi: 10.12928/telkomnika.v18i2.14106.

BIOGRAPHIES OF AUTHORS



Marta Botella-Campos was born in Valencia, Spain, in 1982. She received her Bachelor's degree in Telecommunications Systems, Sound and Image Engineering in 2019 and joined the research institute Instituto de Investigación para la Gestión Integrada de zonas Costeras (IGIC), at the Universitat Politècnica de València, short after finishing her studies. Currently, she is studying a master's degree on Technologies, Systems and Communications Networks at the Universitat Politècnica de València. Email: marbocam@teleco.upv.es



José Luis García-Navas was born in Murcia, Spain. In 2018 he finished his bachelor's degree in Telecommunication Technologies and Services Engineering at the Universitat Politècnica de València (UPV) and he finished a postgraduate Master in Telecommunication in 2020. Since 2019, he is a research member of the Instituto de Investigación para la Gestión Integrada de zonas Costeras (IGIC), at the UPV. He is currently PhD student in telecommunications. His research interest includes on the Internet of things and the application of sensors, on routing protocols and networking, and on wireless technologies. Email: jogarna3@teleco.upv.es



Albert Rego was born in Valencia, Spain on December 13, 1991. He received his bachelor's degree in Computer Science and in Telecommunications Technology Engineering in 2015. In 2016, he received a M.Sc. in Telecommunications by the Polytechnic University of Valencia. He is currently working in his PhD under the FPU national scholarship. His researched lines are focused on software defined networks. He is author of several papers and has cooperated in some international conferences, both by reviewing papers and being a part of a committee. Email: alremae@teleco.upv.es



Sandra Sendra received her degree of Technical Engineering in Telecommunications in 2007. She received her M.Sc. of Electronic Systems Engineering in 2009 and her Ph.D. in Electronic Engineering (Dr. Ing.) in 2013. Currently, she is associate professor at the Polytechnic University of Valencia (Spain). She is Cisco Certified Network Associate Instructor since 2009 and HP-ATA instructor since 2015. She is vocal inside the IEEE Spain Section for the term 2020-2021, member of the Membership Development group for the term 2018-2019 and active member inside the IEEE WIE Spain for the term 2016-2018. She has authored 6 book chapters and 2 books. She has more than 110 research papers published in national and international conferences, international journals (more than 45 with ISI Thomson JCR). She has been the co-editor of 8 conference proceedings and guest editor of several international journals. She is guest editor several SI in International Journals related to underwater communications, sensors, and actuator networks. She has been the co-editor of 9 conference proceedings and associate editor in 6 international journals. She has been involved in more than 100 Program committees of international conferences, and more than 50 organization and steering committees. She has been the general chair (or co-chair) of 5 International conferences. Her research interests, but no limited, include saving energy techniques in electronic circuits, sensor deployment, WSN, UWSN and the application of these technologies for environmental monitoring. Email: ssendra@ugr.es



Jaime Lloret received his B.Sc.+M.Sc. in Physics in 1997, his B.Sc.+M.Sc. in electronic Engineering in 2003 and his Ph.D. in telecommunication engineering (Dr. Ing.) in 2006. He is currently Associate Professor in the Polytechnic University of Valencia. He is the Chair of the Integrated Management Coastal Research Institute (IGIC) since January 2017 and he is the head of the "Active and collaborative techniques and use of technologic resources in the education (EITACURTE)" Innovation Group. He has been Internet Technical Committee chair (IEEE Communications Society and Internet society) for the term 2013-2015. He has authored 14 books and has more than 600 research papers published in national and international conferences, international journals (more than 300 with ISI Thomson JCR). He has been the co-editor of 54 conference proceedings and guest editor of several international books and journals. He is editor-in-chief of the "Ad Hoc and Sensor Wireless Networks" (with ISI Thomson Impact Factor), the international journal "Networks Protocols and Algorithms". Moreover, he is Associate editor of "Sensors" in the Section sensor Networks, he is advisory board member of the "International Journal of Distributed Sensor Networks" (both with ISI Thomson Impact factor), and he is IARIA Journals Board Chair (8 Journals). Furthermore, he is (or has been) associate editor of 46 international journals (16 of them with ISI Thomson Impact Factor). He has been involved in more than 500 Program committees of international conferences, and more than 160 organization and steering committees. He has led many local, regional, national and European projects. He was the chair of the Working Group of the Standard IEEE 1907.1 from 2013 to 2018. Since 2016 till today he is the Spanish researcher with highest h-index in the TELECOMMUNICATIONS journal list according to Clarivate Analytics Ranking. Moreover, he is included in the world's top 2% scientists according to the Stanford University List. He has been general chair (or co-chair) of 63 International workshops and conferences. He is IEEE Senior, ACM Senior and IARIA Fellow. Email: jlloret@dcop.upv.es