Journal of Real-Time Image Processing (2023) 20:100
https://doi.org/10.1007/s11554-023-01357-w

RESEARCH

A face recognition application for Alzheimer's patients using ESP32-CAM and Raspberry Pi

Thair A. Kadhim[1] · Walid Hariri[2] · Nadia Smaoui Zghal[3] · Dalenda Ben Aissa[1]

Received: 24 March 2023 / Accepted: 18 August 2023 / Published online: 28 August 2023

Abstract

This paper proposes a real-time face recognition application to aid people living with Alzheimer's in identifying the people around them. This is achieved by developing a portable system consisting of glasses with an ESP32-CAM and a single-board microcomputer (the Raspberry Pi). The proposed system operates automatically and does not require physical interaction with the user. It utilizes wireless technologies to capture real-time video frames of human faces and transmit them (via Wi-Fi) to the Raspberry Pi, which detects and recognizes the captured human face and sends voice-activated feedback to the user's ears over Bluetooth to pronounce their name. Several incompatibility challenges are encountered and appropriately handled during the system's development, integration, and testing processes. A fully functional prototype is developed and tested successfully. When compared to the state-of-the-art, the obtained results have demonstrated superior performance in terms of a training accuracy of 99.46% and a face recognition accuracy of 99.48%. The entire processing time from capturing the human face to generating the voice message is found to be about one second (730 ms on a laptop and 1109 ms on a Raspberry Pi). The developed technology is anticipated to improve the patient's quality of life and reduce their dependence on others.

Keywords Alzheimer · Assistive technology · ESP32-CAM · Face recognition · Raspberry Pi

1 Introduction

A facial recognition system is a tool that allows you to identify and recognize a person in real time based on their facial features. This technology is extremely essential and is utilized in a wide variety of applications for a wide variety of objectives. Therefore, a face recognition system can be used to identify people in open or restricted locations for things like access control, picture matching, identity verification, and other functions, among others [1]. Another use of this technology is to track and identify suspects in real time [2]. Computer vision encompasses all facial recognition methods. All technologies for facial recognition can be included in the field of computer vision. The science and set of tools that enable a device to process and analyze images of the actual world. A process comparable to what the human eye and intellect perform together. Several computer vision techniques are currently implemented in surveillance systems. One of its advantages is that it eliminates the need for continuous human monitoring of images. Given that the algorithm can be programmed to display only pertinent information [3].

In recent years, FR technology has made significant strides thanks to the use of deep learning, notably the convolutional neural network (CNN). The greatest results in recent years have been achieved using the CNN architecture in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) for face recognition [4]. There are several methods for using the CNN. Building a new model from scratch is the first step. Specifically, the architecture of the previously trained model is used, and the model is trained using the dataset. Second, transfer learning using characteristics from a pre-trained CNN is employed in situations when the dataset is large. Transfer learning allows CNNs to be used, where the convolutional base is kept unchanged and the outputs are fed into a classifier. The pre-trained model is used as

✉ Thair A. Kadhim
altaeeth@gmail.com

1 Microwave Electronics Research Laboratory, Faculty of Sciences of Tunis, Tunis El-Manar University, Tunis, Tunisia

2 Labged Laboratory, Computer Science Department, Badji Mokhtar Annaba University, Annaba, Algeria

3 Control and Energy Management Laboratory (CEM Lab), ENIS Sfax, Tunis, Tunisia

a fixed feature extraction approach when the dataset is small or the problem to be categorized is comparable to the one being categorized [5].

Face recognition often has to deal with unpredictable lighting, significant position fluctuations, facial emotions, cosmetics, thinning or growing facial hair, aging, and partial occlusions [6]. Similar to this, movies are often captured from moving cameras or in unpredictable circumstances in surveillance settings [7]. The performance of face recognition as well as other aspects that might affect matching scores can really be considerably impacted by a number of difficulties and important variables. Several of these difficulties include structural attributes such as facial hair, eyewear, and other components, in addition to primary issues that are currently affecting facial recognition algorithms, such as age, illumination, and pose. Moreover, Hariri et al. [8] classified a typical facial recognition system into the following modules: (1) a sensor module collects two-dimensional and three-dimensional data about the face; (2) a module for feature extraction is used to process the sensor's output in order to extract a collection of discriminating features. (3) Matching module: The extracted characteristics are compared to the stored scans, and matching scores between faces are produced in the matching module. Matching scores are used to determine whether a face can be recognized or not.

Each facial recognition system operates in two steps. Enrollment is the initial step, during which a person utilizes the system for the first time. During the enrolling step, the system stores the user's face information in a database named Gallery. In the second step, the system classifies all faces in the gallery according to their resemblance to the probe. Generally, people visit the elderly every week since there are a lot of family members and temporary personnel caring for them. As a result of Alzheimer's patients' memory loss, they may find themselves in potentially humiliating circumstances because they cannot recall people's names, faces, or other personal information. They cannot even get in touch with visiting relatives or family members who live nearby. Seniors are left feeling confused and insecure as a result of this. With loss of memory, it is impossible to determine if the visitor is a representative of the elderly care facility or not, and hence it is difficult to identify them. Aging and temporary personnel are two factors that contribute to the difficulty of identifying family members, relatives, and visitors. As people get older, their memories get worse, making it hard for them to remember [9].

This study suggests a real-time facial recognition application to assist Alzheimer's sufferers in recognizing those who live nearby. This is accomplished by creating a portable system made up of glasses equipped with an ESP32-CAM and a Raspberry Pi single-board computer. In contrast to the analogous apps that have previously been published, the suggested system runs automatically and does not ask for any direct user engagement. After receiving real-time video frames from the ESP32-CAM in the eyewear, the Raspberry Pi recognizes the live human face and transmits voice-activated feedback to the user's ears over Bluetooth. Throughout the system's development, integration, and testing phases, a number of incompatibility concerns are discovered and properly addressed. Such technological help for Alzheimer's sufferers is anticipated to enhance their quality of life and decrease their dependence on others.

The remainder of this paper is organized as follows. Section 2 presents a literature review for the previously reported work relevant to the proposed work. An overview of the proposed applications is provided in Sect. 3. The different phases of the work methodology including the dataset used in this study are presented in Sect. 4 describes the work methodology including the dataset used in this study. Section 5 presents and discusses the observations and findings afterwards. The work is then concluded in Sect. 6.

2 Related works

Many studies are being conducted to aid Alzheimer's patients by utilizing technology tools, whether software or hardware. In order to get a better idea of what previous studies have added to this field, this research put these studies into three groups together: face recognition, Raspberry Pi, and ESP32-CAM, which are the main pillars of this research.

2.1 Face recognition

A biometric method called facial recognition identifies a person by their face [10]. All of the facial recognition technologies may be categorized under the umbrella term "computer vision." Image processing is a science that includes a collection of tools that enable a device to analyze and evaluate real-world photographs. A mechanism that is comparable to what the human eye does in combination with the brain is described here. Several computer vision techniques are now being implemented. One of its advantages is that it eliminates the need for human intervention in the continual deal with images. Because it is easy to configure the algorithm to display just the information that is relevant to the situation. Sanchez-Moreno et al. [11] proposed a real-time, unconstrained face recognition system that combined deep learning algorithms. The system used moderate hardware and could operate in a variety of environments. Face detection and recognition are the two primary tasks performed by a facial recognition system. When it comes to face detection, the proposed scheme makes use of the YOLO-Face method. When it comes to recognition, the proposed scheme makes use of FaceNet in conjunction with classification using a supervised learning technique such as the SVM. The face detector achieved an accuracy of more than 89.6% when used with the Honda/UCSD dataset, which operates at a frame rate of 26 frames per second and carries out classification tasks by using darknet-53 images at VGA resolution.

A wearable facial recognition system, such as a smartwatch, can be used for this purpose. People who are blind or have limited eyesight can benefit from the application. The smartwatch's ability to identify people in its immediate surroundings helps the visually impaired. Using a Samsung Galaxy GEAR camera, a K-NN algorithm for facial recognition, and a voice recorder for feedback, this facial recognition device can identify a person's face. Nearly eighty-three percent of the system's output is accurate [9, 12]. In Zhang [13], an AD patient's capacity to identify their closest family members' and nurses' faces, which greatly affects their quality of life, is modelled using computer simulations. Research in this field is aimed at determining how far the disease has progressed and how much cognitive capacity may be recovered. This study's goal is to retrain the brains of patients with Alzheimer's disease (AD) in order to restore their capacity to recognize family members and medical professionals through the use of computers.

Aljojo et al. [14] proposed an Alzheimer's Assistant application, as it's known in Saudi Arabia, designed to help patients stay more autonomous and aid them in their everyday lives. In the case that a patient is lost and has to be located, a tracking bracelet can be used to locate the patient's whereabouts. Face recognition and a GPS bracelet will be used for tracking purposes in the proposed application. Mobile Application Timeless [15] is the first book of its type to focus on Alzheimer's sufferers. This is basic, user-friendly software for the elderly. Using the app helps them recall events, stay in touch with their loved ones, and identify people they meet. As a result, Alzheimer's sufferers' loved ones may stay in touch with them. A number of nations, including the United States, Hong Kong, Singapore, Canada, and Japan, are presently able to download the app on their iPhones or iPads. Iraq, on the other hand, has no access to it. Face recognition based on artificial intelligence is used to assist patients in recognizing their loved ones.

2.2 Raspberry Pi

Salman and Rasheed [16] proposed a face recognition system based on the Raspberry Pi that utilizes standard face detection and identification methods. An essential part of maintaining public safety and security is using facial recognition technology. It is referred to as a biometric approach since it is used to identify facial photos by analyzing the fundamental composition of the face. Wazwaz et al. [17] created a system that consists of a collection of computers linked to a microcomputer equipped with a camera. The system captures photos of individuals and uses image processing techniques to analyze, identify, and recognize human faces. The device has the potential to improve safety in crowded places like shopping centers, campuses, and airports. It can recognize and identify facial features in a variety of conditions and scenarios. This system detects human faces using the "boosted cascade of simple features". The LBP algorithm is used to identify these individuals. The Raspberry Pi serves as the brains of the operation, communicating with a camera so that images may be taken. Raju and Srinivasa Rao [18] are creating a facial recognition system using the Raspberry Pi and conventional face detection and identification methods such as a Haar cascade classifier for detection and an LBP feature extraction algorithm for feature extraction. They used the Raspberry Pi II to design a technique for recognizing expressions in real-world scenarios based on geometric distinctive characteristics. They achieved an overall accuracy of 94% using a Raspberry Pi II and an average processing time of 120 ms in a Linux environment (ARM1176JZF, 900 MHz).

Lee et al. [19] investigate the possibility of constructing a face recognition system based on the Raspberry Pi utilizing standard face detection and recognition methods such as Haar detection and principal component analysis (PCA). They are developing facial recognition technology to the point that it may be used in lieu of passwords and radio frequency identification cards for access to high-security systems and buildings. Umm-E-Laila et al. [20] provide the architecture and detailed design for a comparative examination of three alternative implementations of a real-time face recognition system employing local binary patterns (LBP) histograms, Eigen faces, and Fisher faces. These algorithms were selected because they are accurate, and their speed factor may be enhanced by using suitable image analysis gear, such as a Raspberry Pi with an HD camera. The graphical user interface system was developed to learn, identify, and recognize the user in real-time utilizing camera data as input, as well as to indicate the time taken by each function. The paper compares all three methods with various system setups. It has been noted that LBPH takes longer than the other algorithms. These results indicate the LBPH algorithm's supremacy over others and highlight that even on a mediocre machine, this algorithm produces the best results possible due to the utilization of the Raspberry Pi.

Wankhede et al. [21] developed a criminal identification system based on the Raspberry Pi that focuses on automated monitoring with the goal of recognizing individuals on a watch list. This system utilizes an automated surveillance camera to do real-time facial recognition. The suggested method consists of four steps: (1) real-time image training (2) face detection using the Haar classifier (3) examination of trained versus real-time images with security camera images (4) A conclusion based on comparison: the primary program used in this research is OpenCV (an open-source computer vision program). To identify faces, the system employs a variety of techniques, including the Haar cascade, linear SVM, and deep neural network. The primary strategy presented in this study is that when a person approaches the Pi camera, it will first look for probable matches that have previously been saved in this system. If the module detects a match, it captures the individual and notifies headquarters.

2.3 ESP32-CAM module

Jadli et al. [22] presented a project to monitor and identify people who are hiding their identities behind masks using an ESP32-CAM. This project was to address the repercussions that occurred during the COVID-19 pandemic. The project is keeping an eye on everything from the cloud and broadcasting the live feed on the server. To determine whether an individual is concealing their identity using a mask, a convolutional neural network is constructed. It runs on a home-built server and can be tracked from afar. A face recognition system for smart door locks was proposed by Adi and Wahyu [23] using the ESP32 camera. The ATmega328 tiny microcontroller, Arduino board, and FTDI programmer are all used in this. Allafi and Iqbal [24] provide instructions for integrating an ESP32-based web server into a microgrid-scale PV power system to track and record information about the PV and batteries' current and voltage. The system is built around inexpensive components, including sensors, an ESP32 microcontroller, Wi-Fi, and an SD card reader. The ESP32 acts as a sensor data collector. There is a connection between the ESP32's SPI pins and an SD card, where all of the data is stored in text format. After a week or more, the system overwrites the text file on the SD card with fresh information. The SD card is where the website's file resides.

3 System overview

The use of deep learning algorithms, in particular CNNs, to issues pertaining to computer vision has made significant strides in recent years. Because of this, today's vision systems are very dependable, efficient, and adaptable. In order to train each successive layer of a CNN, it must first receive a huge quantity of labelled input data. The output of the CNN is a predicted class, or face identification in the case of facial recognition, together with a confidence value. The classification predictions of such networks are extremely accurate. In order to use the CNN's prediction output in a facial recognition scenario, the network must be trained on all the individuals to be recognized. This requires a dataset containing numerous images for each authorized individual. These network architectures are advantageous in terms of both efficiency and effectiveness [25]. This study used mobileNetV2 for detection and denseNet-161 for classification. The used equipment, its characteristics and features, and the programs it runs are analyzed holistically. It may be easy to recognize a face in a likely image, but it is a different story for a computer. The computer decides which parts of the image belong to the face and which do not. The study begins with the design of a prototype that contains an ESP32-CAM between the two lenses. This camera has Wi-Fi to send streaming video to the Raspberry Pi device, which, in turn, processes this video, detects the faces in the frames, recognizes them, and pronounces the name of the owner of the destination. Furthermore, the research presents the design and architecture of an application utilizing deep learning with this equipment. Figure 1 illustrates the diagram of the system.

However, numerous Internet of Things implementations make use of the ESP32-CAM. It may be used for wireless positioning system signals, wireless identification using QR codes, wireless monitoring in the workplace, and other Internet of Things uses. IoT applications will find it to be the perfect option. Furthermore, the Software Specifications are: (1) a website developed using HTML, CSS, and JS; (2) a server running on Django; (3) Open CV for image processing; (4) CNN for feature extraction and model creation using TensorFlow and Keras API; and (5) Arduino IDE/ESP-IDF for SoC programming [22]. The most fundamental protocol for client–server web architecture is the HTTP protocol, but the problem with the HTTP protocol is that whenever we communicate over it, the client must repeatedly request the server for data, which is not suitable for streaming, which requires a protocol that does not need to repeatedly request

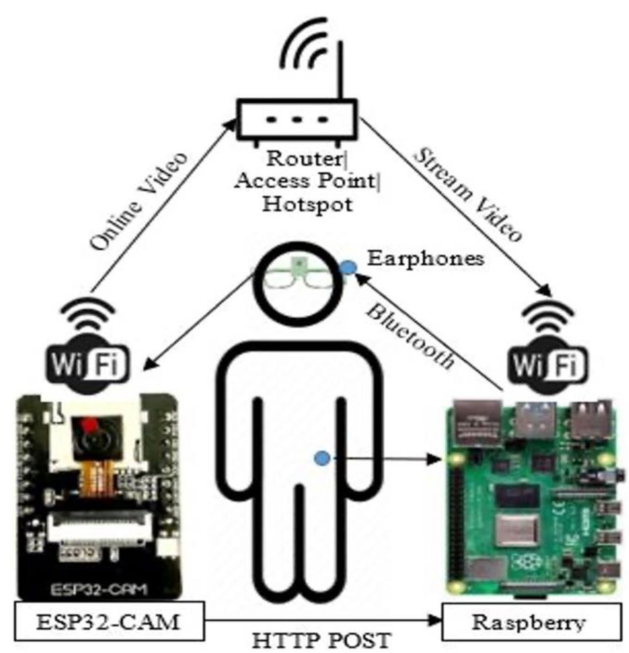


Fig. 1 Schematic of the proposed system

data. Instead, we want a protocol that continuously transmits data across a link without requiring repeated requests [22]. The researchers in this study used HTTP streaming, a solution to this problem in which the server is configured to hold onto a client's request and keep the response open so that data may be sent over it. Updates to the request are sent back across the request-response channel when the server has them, and the connection is closed only when specifically told to.

In this model, without the overhead of HTTP headers and connection opening/closing, a client may simply wait for updates from the server and get them. Furthermore, the study implements the training of face detection and recognition in Google COLAB (GPU). It used Google COLAB Pro+, an NVIDIA Tesla V100 GPU, 53 GB of RAM, and 8 CPU cores. The study trains the MobileNetV2 CNN using the wider dataset. Whereas the model that was trained to recognize faces used the FERET dataset, this research employs transfer learning, a popular approach. It's a method that involves transferring the weights or information gained from one problem's solution to another that's closely similar. Transfer learning for convolutional neural networks is the process of training a model on two or more datasets while retaining the learned features (neuron weights) from the prior dataset [28]. According to the degree of similarity between the datasets, it enables the subsequent dataset to achieve better results with less training [50].

4 Methodology

Generally, there are two main applications for facial recognition technology. The initial one ensures verification, which compares the current image to a previously requested face. The second method is identification, which compares the present image to numerous others stored in a database in order to assess their degree of resemblance. The application built in this study employs the second scenario, in which the image is evaluated alongside all system users. The purpose of this work is to construct a real-time face recognition system that operates on the Raspberry Pi 4 Model B (8 GB RAM) hardware and the Raspbian operating system. Essentially, the Raspberry Pi receives images from the prototype, which has an ESP32-CAM. These images are received and analyzed during transmission, allowing for the recognition and identification of any person within the camera's range. The developed system has two significant characteristics: the first is that it operates in real time, and the second is that it makes use of a face recognition model with a high degree of accuracy. Likewise, the suggested system design is used to build a real-time face recognition system for patients with Alzheimer's disease by utilizing software and hardware. The technique begins by populating the dataset with photos of numerous individuals. The system architecture incorporates ESP32 camera modules in glasses that will be used to generate an immediate video stream and send it to the Raspberry Pi through Wi-Fi technology, as shown in Fig. 2.

The Raspberry Pi reads the output from the camera directly and collects it into frames. The Raspberry Pi will identify faces in collected frames using a MobileNet model. A camera's output picture (of sufficient resolution) is the starting point for face detection, which may then be generated automatically using the Python Integrated Development Environment. and transmit them to the "facial recognizer algorithm" for recognition. Identifying the face in the image and determining if it corresponds to any of the image sets included in the folder. Due to the positive results observed in prior experiments, a DenseNet-161 classifier was employed. The preprocessing step is the first one in the face recognition process. After that comes the face representation step, which is followed by the face matching with the database step. After that comes the face classification step. Finally, the name is spoken. The Raspberry device sends the sound of a name to a wireless earbud in the patient's ear using Bluetooth technology. Figure 3 depicts the phases of development methodology.

Raspberry Pi, an embedded device popular among academics, is often used because to its adaptability, cheap cost, and flexible programming options. The weight, size, and high power consumption of PCs are some of the issues that can be mitigated with the help of Raspberry Pi [26]. According to Syafeeza et al. [27], when they implemented the usage

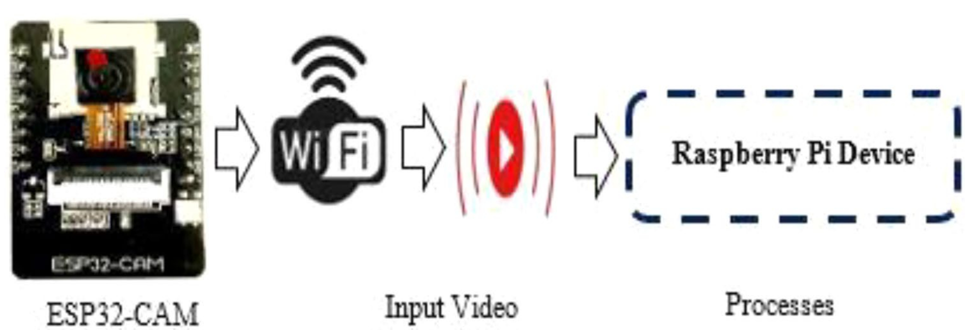


Fig. 2 Real-time and input video source

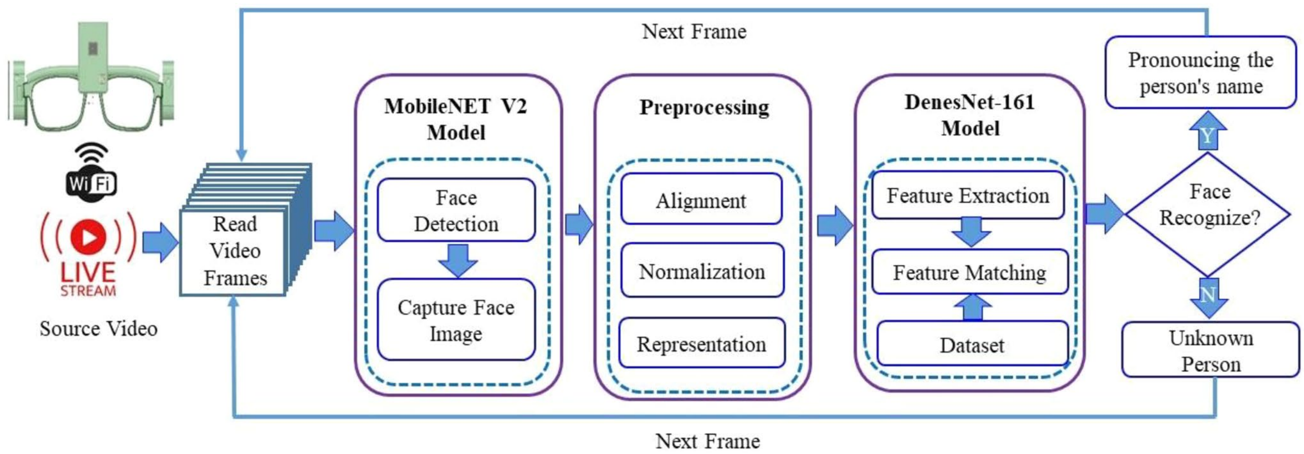


Fig. 3 Phases of development methodology

of Raspberry Pi for image capture system, the system shrank in size, lost weight, and used less energy. Compare that to a PC-based facial recognition system, and you will see that this one is far more practical. Through the use of the Raspberry Pi kit, the framework of this study makes it easy to use, lower cost, and high-performing at the same time. The relationship between speed and accuracy in facial recognition systems, one is accomplished at the expense of the other. As part of this research, a face detection and identification system that is capable of processing images very quickly while also achieving very high true positive face detection rates will be developed. When it comes to hardware implementation, the Raspberry Pi and Python are used in the suggested system. The use of the Raspberry Pi eliminates the need for the underlying hardware platform (CPU, RAM, and operating system) to function properly.

The Raspberry Pi device is a low-cost minicomputer, the size of the Raspberry Pi card is 88 × 58 × 19.5 mm, and the weight is 45 g. This solves the limitations of computers such as size, weight, and power consumption. The weight of ESP32-CAM is 10 g. The total weight of glass parts is 80 g, including polylactic acid (PLA) 3D design, ESP32-CAM, IC programmer, Li-Polymer battery and charging board. The total cost of materials for the prototype is about 120 dollars. This study is used Raspberry Pi 4 Model B with 8 GB of RAM. It is an upgrade over the Raspberry Pi 3 Model B+ in terms of processing speed, multimedia prowess, memory, and connectivity, while still being compatible with older peripherals and using less power. The Raspberry Pi 4 Model B's desktop performance is on par with that of low-end × 86 PCs. Among its many impressive features are a 64-bit 1.5 GHz quad-core processor, dual-display support at resolutions up to 4 K via a pair of micro-HDMI ports, hardware video decoding at up to 4Kp60, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0. It is operational between freezing point and boiling point and 5 V DC through a USB-C connector (minimum 3A*).

The other part is a wireless earbud with a charging case. The transmission of sound through Bluetooth is an example of true wireless technology. Wireless communication is enabled by a very small chip that is inserted into both the sending and receiving pieces of equipment. This technology enables users to hear authentic stereo sound quality without the need for cables or connections of any kind. The Bluetooth standard of this part is V5.0, and the battery type is Li-polymer. The capacity of this battery is 50 mAH (earbuds) and 480 mAH (charging case), which means the charging time is about 3 h. In addition to some steps and instructions on the Raspberry Pi to return the sound produced from the Python application to the earphone via Bluetooth.

4.1 Dataset

Face detection, face localization, and face tracking are all distinct ways of looking at the same concept. Simply put, "facial detection" refers to the process of identifying a human face in a picture. Eye, nose, and eyebrow detection in still images is called "facial localization," while "face tracking" refers to the same process applied to moving images [28]. In line with Yang et al. [29], this study is used WIDER Face dataset which represents the most popular and widely used face detection benchmark, was launched. WIDER Images of faces were gathered from several search engines for LSCOM defined event categories, then manually filtered to get rid of duplicates and those lacking faces. Faces varying greatly in size, position, and occlusion are represented in the WIDER FACE dataset's 32,203 photos and descriptions. There are three distinct sets of data in the database: the training set (40%), the validation set (10%), and the testing set (50%).

In addition, the photos are classified as Easy, Medium, or Difficult, depending on how challenging the detection was.

Furthermore, the FERET dataset was used in this work to train the facial recognition data. The collection of faces is enormous and has been divided into two categories: seclusion and development. Researchers get access to FERET dataset in the development section, whereas the piece that is kept secret is utilized for testing face recognition systems [30]. It is used for a number of purposes, including but not limited to the determination of age, gender, facial recognition, and categorization of individuals. In this particular investigation, a face recognition dataset is used. In this particular investigation, a face recognition dataset is used. May of 2021 was the month when the NIST provided it with the color FERET dataset. Images of a person shot from various perspectives in both color and grayscale are included in this collection. Every image collection has a single label assigned to it that specifically identifies the subject of the image. There are 14,126 total images, representing 1199 individuals in 1564 image sets [31, 32]. Although the FERET samples varied in terms of posture and expression, they were all taken in a controlled environment. Due to the 80:20 divide between the train and test datasets, classes with fewer than five images must be removed from the dataset. Reorganize the class labels as well, since several of the courses were dropped. 994 classes were present before preprocessing, while 555 classes remained after preprocessing. After filtering, there are now 8410 images. The study also added 938 images for eight classes as family and relative of patient with Alzheimer. The total of class is 562 and the number of images has become 9348.

4.2 Face detection

The MobileNetV2 model is used in this research to train its face detection algorithms. MobileNetV2 is an improved version of V1, making it much more powerful and performance-wise efficient [33]. According to Isuyama and Albertini [34], MobileNetV2 is the model that has the least amount of complexity. This is a result that was anticipated, given that MobileNetV2 was developed to be efficient specifically for mobile and embedded vision applications that have limited memory and computational power. Also, they identify that MobileNetV2 is the model with the lowest overall memory usage. Also, it is a small and efficient deep neural network that outperforms other deep neural networks in terms of classification accuracy while using less parameters [35]. A batch size of 32 was used throughout the training process for the MobileNetV2 model, which was run for a total of 25 epochs. The model attained an accuracy of 99.46% during training and 99.65% during validation, with a train loss of 0.0128 and a validation loss of 0.0307. The training accuracy and validation loss across 25 Epochs are presented in Table 1. It is clear from the table that there is hardly any difference between the accuracy of the training data and the validation data, which indicates that the model does not suffer from overfitting. Figure 4 illustrates the architecture of MobileNetV2.

4.3 Preprocessing

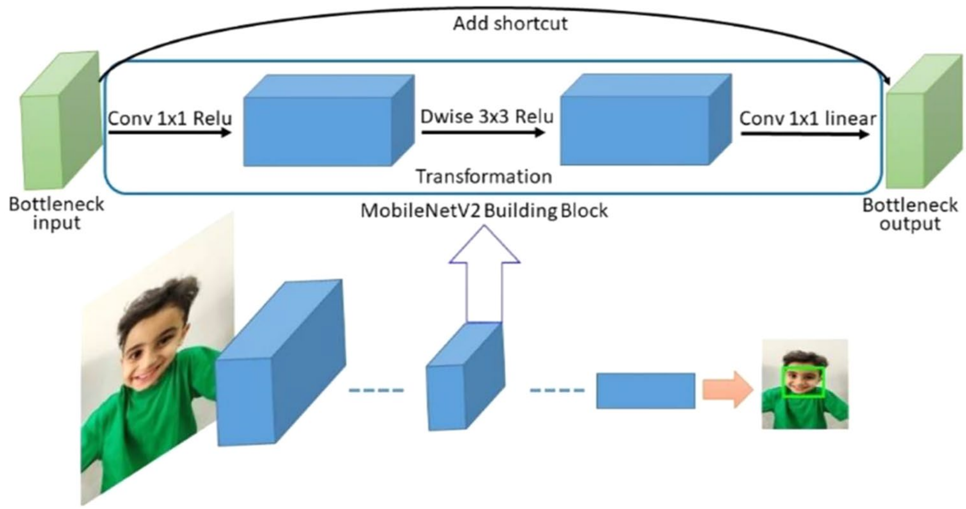
To build models for identifying the individual in an image, a full image is not required for this purpose; simply just the face portion of the image is required moving forward. This produces a collection of face bounding box coordinates for each image. OpenCV is used to do a grayscale conversion on the images as well. The grayscale transformation of the images is required in order for the model to provide better results. Figure 5 illustrates a sample of output preprocessing image. Moreover, the processes involved in facial recognition entail a variety of procedures, the most important of which are the collecting, evaluating, and contrasting of your face with a database of previously saved images. This study increases variance in the dataset and reduces overfitting through data augmentation using the Albumentations package in Python. The augmentations used to resize the input image, rotate the image, vertical flip, apply Gaussian noise. The following is a list of the fundamental procedures that are used by the face recognition system in order to acquire images and compare them:

- *Alignment*: After the facial landmarks have been detected, the algorithm calculates the head's orientation, size, and posture. A minimum of 40° of head movement toward the camera is required for the system to recognize the face.

Table 1 The output details of training

Details	Detection	Recognition
Dataset	Wider	FERET and patient's dataset
Number of classes	32,203	562
Number of images	393,703	9348
Epoch	23	25
Total parameters	3,997,200	27,697,995
Trainable parameters	3,958,824	27,697,995
Non-trainable parameters	38,376	None
Training time	5361 s	3464 s
Number of training images	314,962	7478
Number of testing images	78,741	1870
Image resolution	512*512	224*224
Algorithm	CNN	CNN
Technique	MobileNetV2	DenseNet-161
Accuracy	0.9946	0.9948

Fig. 4 MobileNetV2 architecture



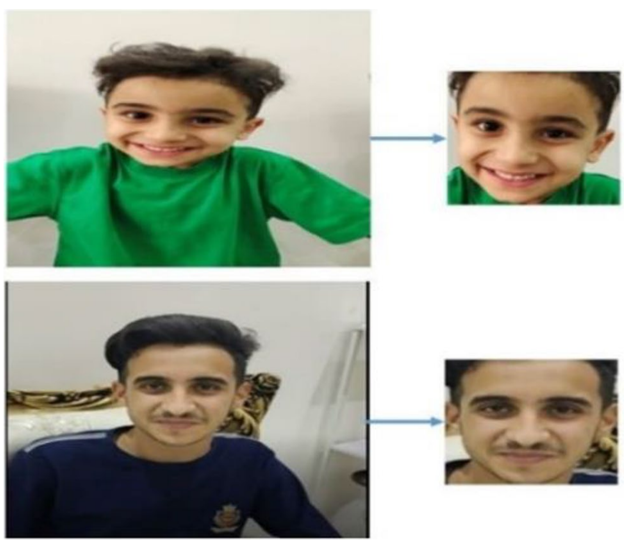


Fig. 5 Example of a face cropped from an original image

- *Normalization*: It is necessary to scale and rotate the head picture before it can be properly registered and mapped. No matter where the head is or how far away it is from the camera. The normalizing procedure is unaffected by ambient light.
- *Representation*: After the face data has been normalized, the system assigns it a unique code. Coding makes it possible to more clearly express and compare freshly obtained face data to previously stored facial data.

4.4 Face recognition

This study is used DenseNet-161 model for face recognition which it get best accuracy in our previous study [36]. As its source of input data, each layer of this model is given the feature maps corresponding to all of the expected labels. The output from this layer is used by the subsequent layers to calculate their own findings [37]. In order to achieve its objectives as efficiently as possible, DenseNet-161 prioritizes the use of shortcut connections. When using this approach, each layer is directly linked to the other layers in the structure. In addition, the last layer of the model was modified throughout the research to reduce the number of neurons from 1000 to 562, which corresponds to our number classes. A maximum of 25 epochs were used to train the DenseNet model with a batch size of 32. With a train loss of 0.0128 and a validation loss of 0.0307, the model was able to achieve a 99.48% accuracy in training. Information and results from training two models are shown in Table 1. Convolution, pooling, dense block (DK), transition layer (TL), and fully connection layer (FL) are shown in Fig. 6, which is an illustration of the architecture of the DenseNet-161 layers.

5 Results and discussion

5.1 Experimental results

The greatest obstacle to face detection and recognition applications is the speed of response in real time. The process of identifying and differentiating faces requires a device capable of rapid processing, such as servers, whether local, cloud, or personal. The time necessary to reach these servers is what causes the quick facial recognition capability to be lost. This research sped up the process by connecting the camera in the glasses directly to the Raspberry Pi, which is how the whole process is done. When processed via recognition, all identified faces are assigned a category. A response is sent upon completion of this procedure, which may include either the user's name (if known) or a message "unknown" if no match is found in the dataset. An example of system response to unknown person is shown in Fig. 7.

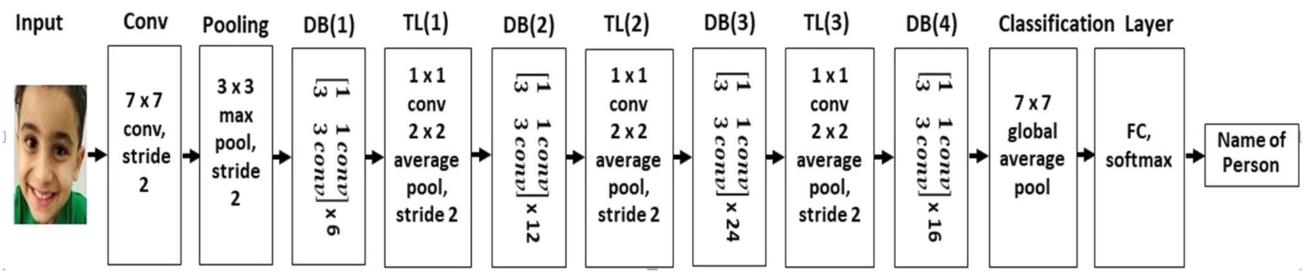
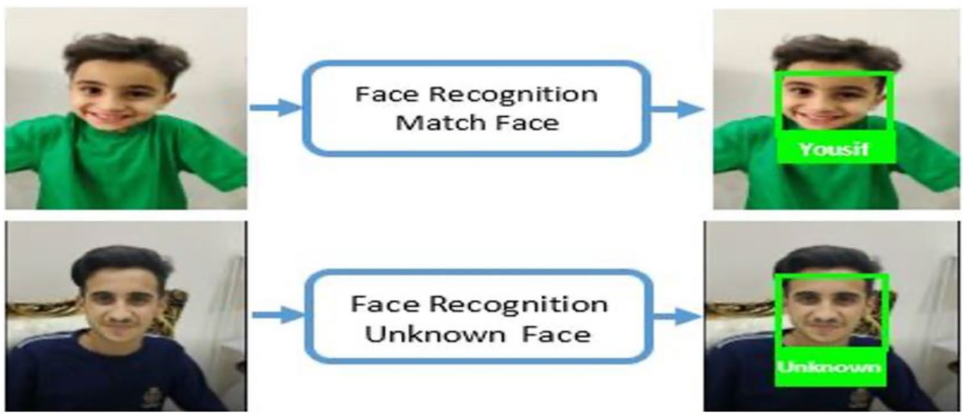


Fig. 6 Example of DenseNet-161 architecture

Fig. 7 An example of system response to unknown person



An experiment in which the laptop and Raspberry Pi received video frames from the ESP32-CAM via Wi-Fi is initially carried out. In this experiment, several data formats are used such as .jpg, .bmp, .mjpeg. Each type format has its pros and cons in terms of size, speed, and image quality. The image is received and the success of the sending and receiving processes between the equipment used in this study. This study conducted several experiments based on image type (.bmp,.jpg) and image resolution to obtain a good image quality in a reasonable amount of time. Bitmap is a raster-based file format that was developed in the early days of computer graphics to show images in a manner that was independent of the devices being used to display them. Because BMP files include a significant amount of information, their typical sizes are rather huge. JPEG stands for "Joint Photographic Experts Group," and it is a raster-based image format that has gained widespread use. JPEGs find a compromise between lowering file sizes and keeping image quality thanks to their revolutionary lossy compression technology. JPEGs are built around this method.

Both of these type of files are raster-based, which is a format that has its roots in the early days of computer graphics and digital photography. Each also displays photos of a high quality that are capable of being compressed. Both BMP and JPEG are capable of maintaining image quality, displaying color, and compressing image data; however, the primary distinction between the two lies in how each format is commonly put to use. The time of image received from ESP32-CAM is different from image to other depend the size of this image. The size of the bmp image was very high compared with the jpg image; for example, the size of the image resolution 800×600.bmp is 1,440,054 bytes, while the image received with the same resolution, 800×600.jpg, is 21,424 bytes.

The next experiments are on a laptop with a Core i5, 8 GB of RAM, an SSD hard disk, and a Windows 10 64-bit operating system. This implementation uses a ESP32-CAM to send video frames to the application that contain some faces, some of which are in the dataset and others of which are not. The time between receiving the video and splitting it into frames for detecting and recognizing the face and pronouncing the name ranged between 442 and 979 ms for 5 frames per second, 481 and 1232 for 10 frames per second, 515 and 1806 for 15 frames per second, and 564 and 1530 for 20 frames per second. These results indicate that there is an equation between image resolution and the number of frames that are processed per second. The higher the number of frames, the longer the time.

The study also had multiple experiences using the Raspberry Pi camera, which has a suit features for recognizing faces. The camera has the following features and specifications: 5 MP Omnivision 5647 Camera Module, 2592×1944 Still Image Resolution, Video: Recording modes include 1080p at 30 frames per second, 720p at 60 frames per second, and 640 × 480p at 60 or 90 frames per second. The 15-pin MIPI Camera Serial Interface, which measures 20 × 25 × 9 mm, connects directly to the Raspberry Pi Board. The results from the experiments with this camera are appropriate and close to the results of implementing the application on the laptop because the camera is connected to the Raspberry Pi via a ribbon wire. This connection is easy to install, and the camera is fully compatible with the Raspberry Pi device, both as software and hardware. But this type of connection is avoided in this study, because it contradicts the major objectives, which are to have all parts of the study work automatically and to adopt wireless communication to reduce effort and the burden of using technology on the patient. Figure 8 illustrates image output and how to connect the Raspberry Pi with its camera and other equipment like a monitor, keyboard, and mouse.

During the process of making a video, the phrase “frame rate” refers to the pace at which a picture changes from one frame to the next. Another name for this is “frame change rate.” When it comes to evaluating the overall quality of a video, the frame rate is one of the most important factors to consider. The vast majority of studies that have been conducted on the relationship between frame rate and video quality have focused on low frame rates, which are often defined as having less than 30 frames per second (fps) [38]. Furthermore, when the study involves experiments with the Raspberry Pi using ESP32-CAM, it begins with one frame per second and uses several types of resolutions and sizes. The results of this experiment are 990 ms for high resolution 1600 × 1200 and 775 ms for low resolution 160 × 120. In this experiment, one frame per second is used to figure out how long it takes from when the camera type OV2640 sends the video, goes through the receiving and processing steps, and pronounces the person’s name to the patient. Figure 9 shows the average time recorded in this experiment for transferring types of frames.

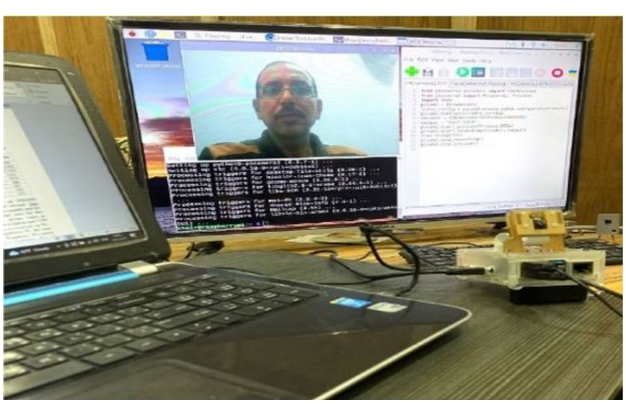


Fig. 8 Raspberry Pi camera connection and output

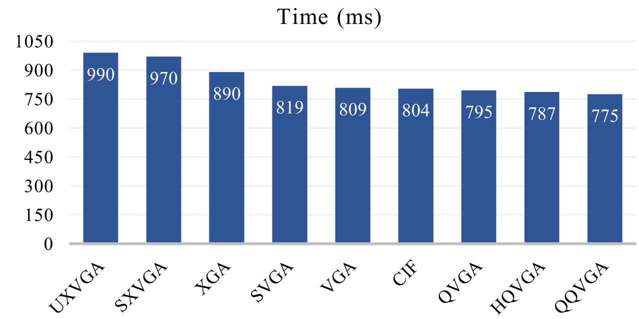


Fig. 9 OV2640 Frame size versus time (ms) for a single FPS in Raspberry Pi

In other experiments, some of the results are found to be different when carrying out the application on the Raspberry Pi using the same ESP32-CAM. As previous experiment the average time from receive video from ESP32-CAM to pronounce the name depend the resolution and number of frame per second. The results obtain using Raspberry Pi are between 807 and 1640 ms for 5 frames per second, 823 and 1478 for 10 frames per second, 824 and 1167 for 15 frames per second, 874 and 1192 for 20 frames per second. The experiment of using an ESP32-CAM, which sends frames of video to a Raspberry Pi device through Wi-Fi, there are several resolutions for cameras, and they yield varying frame rates, measured in both frames per second (FPS) and milliseconds (ms). Table 2 displays the comprehensive comparative findings for the ESP32 in various resolutions in Raspberry Pi.

This study used an SVGA frame size, an 800x600 resolution, and a frame rate of ten frames per second. This option is the middle ground between the image quality and the speed of sending and receiving. The average time for this option to capture a face from an online stream video to pronounce the name is 730 milliseconds on a laptop and 1109 milliseconds on a Raspberry Pi. The video is received via Wi-Fi in experiments on a laptop and a Raspberry Pi; the time delay in implementation is due to the Raspberry Pi’s lower processing power when compared to a computer. Figure 10 shows an example of average SVGA results obtained from the serial monitor of Arduino IDE.

Furthermore, the study examined the accuracy of live images, the number of image classes that were used in the examination was eleven, eight of them in the dataset, and three out of the dataset. The result was that the eight images that were shown in the live images were true, and the three were false. On the other side, there is one image that was not recognized in the live image when the images were in difficult positions and angles. The accuracy of the live images that is found to be between 91 and 100% is calculated from [39].

Table 2 OV2640 comparison between PC and Raspberry Pi performance

Frame size	Max FPS	Size (byte)		5-FPS (ms)		10-FPS (ms)		15-FPS (ms)		20-FPS (ms)	
		Laptop	Rasp	Laptop	Rasp	Laptop	Rasp	Laptop	Rasp	Laptop	Rasp
UXVGA	9	69,306	69,306	979	1640	N/A	N/A	N/A	N/A	N/A	N/A
SXVGA	10	48,918	48,918	801	1332	1005	1478	N/A	N/A	N/A	N/A
XGA	13	32,120	32,120	745	1048	835	1250	N/A	N/A	N/A	N/A
SVGA	26	21,424	21,424	603	1045	730	1109	1148	1167	1152	1192
VGA	23	12,934	12,934	524	911	656	959	877	1002	1021	1024
CIF	52	7954	7954	474	856	536	886	607	972	957	986
QVGA	42	5471	5471	470	844	521	861	570	893	710	895
HQVGA	51	3527	3527	428	832	481	840	506	843	571	850
QQVGA	50	2230	2230	442	807	451	823	464	824	564	874

```
11:12:00.005 -> changeResolution(800,600) success
11:12:00.050 -> capture() success: 800x600 20807b
11:12:00.085 -> changeResolution(800,600) success
11:12:00.117 -> capture() success: 800x600 20901b
11:12:00.117 -> changeResolution(800,600) success
11:12:00.197 -> capture() success: 800x600 20778b
11:12:00.233 -> changeResolution(800,600) success
11:12:00.277 -> capture() success: 800x600 20748b
11:12:00.311 -> changeResolution(800,600) success
11:12:00.357 -> capture() success: 800x600 20762b
11:12:00.403 -> changeResolution(800,600) success
11:12:00.435 -> capture() success: 800x600 20875b
11:12:00.482 -> changeResolution(800,600) success
11:12:00.527 -> capture() success: 800x600 20738b
11:12:00.564 -> changeResolution(800,600) success
11:12:00.596 -> capture() success: 800x600 20783b
11:12:00.634 -> changeResolution(800,600) success
11:12:00.681 -> capture() success: 800x600 20760b
11:12:00.724 -> changeResolution(800,600) success
11:12:00.756 -> capture() success: 800x600 20798b
11:12:00.803 -> changeResolution(800,600) success
11:12:00.849 -> capture() success: 800x600 20743b
11:12:00.883 -> changeResolution(800,600) success
11:12:00.923 -> capture() success: 800x600 20755b
11:12:00.963 -> changeResolution(800,600) success
11:12:00.999 -> capture() success: 800x600 20806b
```

Fig. 10 An example screenshot for the real-time serial monitor, FRAMESIZE_SVG (26 FPS) from Arduino IDE

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (1)$$

where TP = true positive, TN = true negative, FP = false positive, and FN = false negative.

5.2 Discussion

Many people remain living on their own until old age with weak or lost memories, and this introduces new challenges to society. This research can provide valuable help to patients, relatives, and can be extended to recognize friends and caregivers. This can provide additional support to the elderly, especially those suffering from impaired vision, memory loss, or dementia. This study also took into account several important considerations, such as the weight of the equipment, lower production costs, and lower energy use. The study utilized the Raspberry Pi, a diminutive computer in credit card dimensions that is economically priced. The physical dimensions of the Raspberry Pi card are $88 \times 58 \times 19.5$ mm, and its weight is 45 g. The use of this device addresses the constraints of personal computers, such as their physical dimensions, weight, and energy usage. The weight of the ESP32-CAM is 10 g. The aggregate weight of the various components of the glasses, such as the 3D model composed of polylactic acid (PLA), ESP32-CAM, IC programmer, Li-Polymer battery, and charging board, amounts to 80 g. The aggregate cost of acquiring the materials for the prototype amounts to approximately 120 dollars. The power consumption of the parts in the glasses is ESP32-CAM (180 mA), IC programmer (20 mA), and charging chip TP4056 (130 mA). The life of the Li-Polymer battery used in this study was more than 3 h. The study used the formula battery life = battery capacity (mAh)/load current (mA) to calculate battery life.

The Raspberry Pi device used in this study exhibits a significantly lower weight than the mobile device. Conversely, mobile devices are utilized for communicating with server devices or cloud infrastructure to facilitate data processing, as they lack the capacity to perform deep learning-based data processing. Therefore, this mode of communication is susceptible to interruptions, disruptions, and limited network coverage, which may result in time loss. The system proposed in this study utilizes the Raspberry Pi device for processing, thereby eliminating the need for an Internet connection. The ease of use and fully automatic implementation of applications have increased confidence in the use of technology, whether from patients or those who care for them alike. The study uses MobileNetV2 for face detection; it is the model with the lowest complexity and was designed to work well with low-resource vision apps like those found on mobile devices and embedded systems. Also, it has the lowest overall memory usage for running. DenseNet is used for face recognition, where it obtained the best accuracy in a previous study.

Although there are various challenges of face recognition such as ageing and wrinkles factor, facial features, expressions, posing and viewpoint, occlusion and illumination variations, as well as accuracy-related challenges from face appearance and imaging conditions, In the research, it was recommended to better address the difficulties of face recognition in real-world circumstances. In addition, there are many other challenges like different origin and manufacture, the difficulty of compatibility of devices with each other, and the difficulty of compatibility of the necessary programs and libraries used in the application with the operating system of the Raspberry Pi, whether it is 32-bit or 64-bit. Tensorflow library is not supported on 32-bit architecture [40], Raspberry Pi OS 64 bits does not support picamera library and so on. The study adopted a new version of the operating system for the Raspberry Pi called "Bullseye," which supported 64-bit OS architecture and required libraries that took about 41 working hours to install. This study was also able to bypass some defects of the ESP32-CAM in the web serve [41] by relying on a private network and avoiding data packet loss caused by interference, noise, or interference from radio frequency or other wireless devices.

However, the study included multiple aspects such as ESP32-CAM, the Internet of Things represented by the Raspberry Pi, and communications by sending video from the prototype's camera to the Raspberry Pi device. Even though the study used different equipment and deep learning, it came up with a novel application for detecting and recognizing the face using a Raspberry Pi device and achieved an accuracy of 99.46% for face detection and 99.48% for recognition. With the ESP32-CAM, which sends frames of video size 21,424 byte to the Raspberry Pi device through Wi-Fi, the time from capturing the face from the online stream video to pronouncing the name is 730 ms in laptop and 1109 ms in Raspberry Pi. Wi-Fi is used to transfer the video between ESP32-CAM and both laptop and Raspberry Pi in the experiments, with the implementation time varying depending on the device's processing power. The research employed a balance between the two criteria of time and speed and reached outstanding results when compared with previous studies in terms of the accuracy of the results or the time taken for implementation. This study can also be used VGA frame size with 640×480 resolution and 959 ms, but, this study preferred this frame size type SVGA because the Alzheimer's patient's movement is slow and it is possible to benefit from increasing the image resolution and quality to ensure that faces in the image are recognized with higher accuracy. Table 3 shows a comparison of the results of this study with the results of previous studies that used the Raspberry Pi device to detect and distinguish faces.

Furthermore, this study achieved integration and compatibility between the Internet of Things devices that were used. Many cameras were found, but they were not compatible with the Raspberry Pi. The prototype presented in this study proved successful in terms of compatibility, quality, and speed. This study was keen to adopt the highest level of simplification for the Alzheimer's patient, as all devices work automatically without the need for patient intervention. In addition, the study avoided the use of wires because of their impact on the difficulty of connection and use, which may negatively reflect the patient's psyche. The study relied on Wi-Fi technology to transmit data between devices and, at the same time, on Bluetooth to transmit sound from the Raspberry device to the patient's earpiece. The time obtained in this study is better than many of the previous studies and the studies that took less time due to the use of wired connectivity or the use of servers and personal computers, while this study used a microcomputer that can be loaded easily and performs the functions that the study sought to achieve. However, the presented work is still open for further improvements in terms of conducting more research to combine face and voice recognition to create more effective applications to assist the aging population as well as children with special needs who suffer from vision impairment or memory loss.

6 Conclusion

A new face recognition application for Alzheimer's patients has been designed, developed, and tested successfully, using ESP32-CAM and Raspberry Pi. The developed prototype that is based on Industrial Internet of Thing devices and components can detect and recognizing human faces without any physical interaction with the user. This is achieved by utilizing wireless technologies such as Wi-Fi and Bluetooth. The ESP-CAM of the glasses utilizes Wi-Fi to send real-time video frames to the Raspberry Pi which in turn does the necessary face detection and recognition, and then returns a voice-activated output to the user's ears through Bluetooth. Such a wireless interface is expected to be appreciated by the elderly users. Furthermore, all the devices that make developed prototype operate automatically. Overall, the developed technology aid is expected to improve the

Table 3 Comparison between different face detection and recognition studies using Raspberry Pi

Researchers	Time (ms)	Resolution	Technique	Accuracy
Teixeira et al. [3]	7200	No stated	A combination of the HOG + SVM for face detection and Residual Network for face recognition	99.38%
Ahmed and Rasheed [42]	4000	No stated	Using OpenCV, Raspberry Pi, and on an Android app on cell phones. Face recognition system on Raspberry Pi	99.63%
Nikisins et al. [43]	112	320×240	Local binary pattern algorithm	99.33%
Suchitra et al. [44]	120	No stated	ASM, Ada boost	94%
Lu et al. [45]	No stated	No stated	Haar feature-based Cascade Classifiers	85.7%
Gsponer [46]	No stated	No stated	Principal component analysis (PCA)	69%
Kak and Mustafa [47]	500	112×92	DWT with PCA	99.25%
Munir et al. [48]	400	No stated	OpenCV algorithms for offline face recognition	90%
Saputra and Surantha [49]	1472	No stated	local binary pattern histogram	98%
	2221		dlib face_recognition	96%
Amato et al. [25]	2900	No stated	VGGNet SENet_ft ResNet-50_ft	93.44% 98.83% 98.87%
Orna et al. [50]	No stated	500×500	Haar cascade algorithm MobileNet-SSD	88.75%
Umm-E-Laila et al. [20]	13.7–15.4	No stated	Eigen face algorithm	No stated
	14.0–15.7		LBPH algorithm	No stated
	12.8–14.5		Fisher face algorithm	No stated
Gunawan et al. [51]	109,945	1920×1080	Principal Component Analysis (PCA)	90%
	25,081	1280×960		
	5695	640×480		
	1451	320×240		
Vamsi et al. [52]	14.0–15.7	No stated	Local Binary Pattern Histogram	No stated
Hasban et al. [53]	120	No stated	Haar Cascade classifier	56%
Nagpal et al. [54]	2000–3000	1280×720	Haar Cascade classifier for face detection and local binary pattern histogram for facial recognition	60–75%
Singh et al. [55]	No stated	No stated	Haar-Cascade Histogram of oriented gradients	98% 80%
Nadaf et al. [56]	No stated	No stated	Yolo technique Haar cascade classifier technique	83%—96%
Raju and Rao [18]	120	No stated	Haar cascade classifier Local binary pattern (LBP)	94%
Novosel et al. [57]	No stated	200×200	Histogram of oriented gradients (HOG)	95%
Parthornratt et al. [58]	No stated	100×100	Assumption University's Raspberry Pi Customer Counter (AU-PiCC)	90%
Preetha et al. [59]	No stated	No stated	Haar CascadeClassifier Local binary patterns histograms	82%
Salman and Rasheed [16]	2360–2660	No stated	Haar-like features Classifier Cascade	99.63%
Proposed method (Laptop)	603 (5 FPS) 730 (10 FPS) 1148 (15 FPS) 1152 (20 FPS)	800×600	FD: MobileNet V2 FR: DenseNet-161	99.46% 99.48%
Proposed method (Raspberry Pi)	819 (1 FPS) 1045 (5 FPS) 1109 (10 FPS) 1167 (15 FPS) 1192 (20 FPS)	800×600	Face detection: MobileNet V2 Face recognition: DenseNet-161	99.46% 99.48%

quality-of-life of the targeted people and make them less reliant on other people.

The developed application utilized MobileNet that worked well in detecting human faces as well as the DenseNet which successfully recognized the face and generated an appropriated voice-activated feedback to the user. In addition, when compared to the state-of-the art, the obtained results have demonstrated and excellent performance in terms of a training accuracy of 99.46% and a face recognition accuracy of 99.48%. On the client side, the frames are also flawlessly rendered, allowing for the production of the video. SVGA was selected as the frame size for this investigation since it has a greater resolution than the others. The Raspberry Pi device receives the video frames from the ESP32-CAM device with no problems. The entire time from taking a picture of a person to producing the output message is about one second (730 ms on the laptop and 1109 ms on the Raspberry Pi).

Throughout the development process of this application, several technical challenges have been addressed including the incompatibility issues between the standard cameras and the Raspberry Pi operating system. Most of them support Windows and Android operating systems while some others support MacOSX, but not the Raspberry Pi. This problem was addressed by utilizing a newly developed operating system, called Bullseye, for the Raspberry Pi. The incompatibility between the IIOT devices and the computer vision development tools was another challenge that was addressed by developing the necessary software interface and careful setup of the operational parameters. Despite the important contribution of this study towards developing assistive technology for people living with Al-Alzheimer, this work is still open for further developments to address some factors related to the user's environment such as illumination, distance between the user and the camera, camera resolution, angulation, and position of the face. These improvement aspects and others are currently part of the Author's ongoing research.

Acknowledgements The authors would like to express their gratitude to Springer Nature for their interest in our study and for the meticulous review process. We also wish to extend our gratitude and appreciation to the peer reviewers for their time and efforts in reading our article and offering helpful and constructive feedback.

Author contributions The authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the Journal of Real-Time Image Processing. Authorship contributions The specific contributions made by each author are in the three categories below. Category 1 Conception and design of study: TK, WH, NS, DBA; Design a prototype: TK, NS; Devices integration: TK, DBA. Category 2 Drafting the manuscript: TK, NS; revising the manuscript critically for important intellectual content: TK, WH. Category 3 The authors have given their permission for the manuscript to be published in its current form: TK, WH, NS, DBA.

Funding There is no external funding.

Data availability An existing image dataset was made available to this study.

Declarations

Conflict of interest There is no conflict of interest.

Research involving human participants and/or animals This study used existing image datasets with a written permission from the author. The datasets are processed lawfully, fairly and in a transparent manner in accordance with the ethical guidelines at Tunis El-Manar University.

References

1. Lin, Z.H., Li, Y. Z.: Design and Implementation of Classroom Attendance System Based on Video Face Recognition. In: Proc. - 2019 Int. Conf. Intell. Transp. Big Data Smart City, ICITBS 2019, pp 385–388, (2019), doi: https://doi.org/10.1109/ICITBS.2019.00101
2. Chuo, Y. H., Sheu, R. K., Chen, L. C.: Design and implementation of a cross-camera suspect tracking system. In: 2019 Int. Autom. Control Conf. CACS 2019, pp. 1–6, (2019)https://doi.org/10.1109/CACS47674.2019.9024367
3. Teixeira, E.H., Mafra, S.B., Rodrigues, J.J.P.C., Da Silveira, W.A.A.N., Diallo, O.: A review and construction of a real-time facial recognition system. Inst. Nac. Telecomun. (2020). https://doi.org/10.5753/sbcup.2020.11225
4. Almabdy, S., Elrefaei, L.: Deep convolutional neural network-based approaches for face recognition. Appl. Sci. (2019). https://doi.org/10.3390/app9204397
5. Boufenar, C., Kerboua, A., Batouche, M.: Investigation on deep learning for off-line handwritten Arabic character recognition. Cogn. Syst. Res. **50**, 180–195 (2018). https://doi.org/10.1016/j.cogsys.2017.11.002
6. Ho, H.T., Chellappa, R.: Pose-invariant face recognition using Markov random fields. IEEE Trans. Image Process. **22**(4), 1573–1584 (2013). https://doi.org/10.1109/TIP.2012.2233489
7. Phillips, P. J. et al.: Overview of the face recognition grand challenge. In: Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005, vol. I, pp. 947–954, (2005), https://doi.org/10.1109/CVPR.2005.268
8. Hariri, W., Tabia, H., Farah, N., Benouareth, A., Declercq, D.: 3D face recognition using covariance based descriptors. Pattern Recognit. Lett. **78**, 1–7 (2016). https://doi.org/10.1016/j.patrec.2016.03.028
9. Martikainen, K., Said, K.: A facial recognition application for elderly care: caregiver verification and identification. KTH R. Inst. Technol. Sch. Electr. Eng. Comput. Sci., (2018), http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1252357&dswid=5861
10. Bakhshi, Y., Kaur, S., Verma, P.: A study based on various face recognition algorithms. Int. J. Comput. Appl. **129**(13), 16–20 (2015). https://doi.org/10.5120/ijca2015907066
11. Sanchez-Moreno, A.S., Olivares-Mercado, J., Hernandez-Suarez, A., Toscano-Medina, K., Sanchez-Perez, G., Benitez-Garcia, G.: Efficient face recognition system for operating in

unconstrained environments. J. Imaging (2021). https://doi.org/10.3390/jimaging7090161
12 de Sousa-Britto-Neto, L., Maike, V.R.M.L., Koch, F.L., Baranauskas, M.C.C., Rocha, A.D.R., Goldenstein, S.K.: A wearable face recognition system built into a smartwatch and the blind and low vision users. Lect. Notes Bus. Inf. Process. **241**(December), 515–528 (2015). https://doi.org/10.1007/978-3-319-29133-8_25
13. Zhang, Y.: A computational model of quantitatively measuring the Alzheimer's disease progression in face identification. Electron. Sci. Technol. Appl. **6**(1), 29–33 (2019). https://doi.org/10.18686/esta.v6i1.93
14. Aljojo, N., et al.: Alzheimer assistant: a mobile application using machine learning. Rev. Română Inform. Autom. **30**(4), 7–26 (2020). https://doi.org/10.33436/v30i4y202001
15. Timeless, "Timeless," (2019) https://kale-clavichord-7blm.squarespace.com/
16. Salman, H.M., Rasheed, R.T.: Smart door for handicapped people via face recognition and voice command technique. Eng. Technol. J. **39**(1B), 222–230 (2021). https://doi.org/10.30684/etj.v39i1b.1719
17. Wazwaz, A. A., Herbawi, A. O., Teeti, M. J., Hmeed, S. Y.: Raspberry Pi and computers-based face detection and recognition system. In: 2018 4th Int. Conf. Comput. Technol. Appl. ICCTA 2018, pp. 171–174, (2018)https://doi.org/10.1109/CATA.2018.8398677
18. Raju, K., Srinivasa-Rao, Y.: Real time implementation of face recognition system on Raspberry Pi. Int. J. Eng. Technol. **7**(2), 85–89 (2018). https://doi.org/10.14419/ijet.v7i2.17.11564
19. Lee, S. J., Jung, S. B., Kwon, J. W., Hong, S. H.: Face detection and recognition using PCA. In: IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 1, no. December, pp. 84–87, (1999) https://doi.org/10.1109/TENCON.1999.818355
20. Umm-E-Laila, Khan, M. A., Shaikh, M. K., Bin Mazhar, S. A., Mehboob, K.: Comparative analysis for a real time face recognition system using raspberry Pi. In: 2017 IEEE International Conference on Smart Instrumentation, Measurement and Applications, ICSIMA 2017, vol. 2017-Novem, no. November, pp. 1–4. (2018) https://doi.org/10.1109/ICSIMA.2017.8311984
21. Wankhede, V., Mule, G., Londhe, R., Tarale, A., Gome, S.: Criminal face recognition using Raspberry Pi. Int. Res. J. Innov. Eng. Technol. **3**(12), 1–3 (2019)
22. Kaur, A., Jadli, A., Sadhu, A., Goyal, S., Mehra, A., Rahul.: Cloud based surveillance using ESP32 CAM. In: Int. Conf. Intell. Technol. Syst. Serv. Internet Everything, ITSS-IoE 2021, no. April 2022, (2021) https://doi.org/10.1109/ITSS-IoE53029.2021.9615334
23 Dani, P., Adi, P., Wahyu, Y.: Performance evaluation of ESP32 Camera face recognition for various projects. ASCEE, Indones. **02**(June), 1 (2021). https://doi.org/10.31763/iota.v2i1.512
24. Allafi, I., Iqbal, T.: Design and implementation of a low cost web server using ESP32 for real-time photovoltaic system monitoring. In: 2017 IEEE Electr. Power Energy Conf. EPEC 2017, vol. 2017-Octob, no. May 2022, pp. 1–5, (2018) https://doi.org/10.1109/EPEC.2017.8286184
25. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vairo, C.: Facial-based intrusion detection system with deep learning in embedded devices. In: ACM Int. Conf. Proceeding Ser., pp. 64–68, (2018) https://doi.org/10.1145/3290589.3290598
26. Januzaj, Y., Luma, A., Januzaj, Y., Ramaj, V.: Real time access control based on face recognition. November, (2015) https://doi.org/10.15242/iae.iae0615004
27 Syafeeza, A.R., Mohd-Fitri-Alif, M.K., Nursyifaa-Athirah, Y., Jaafar, A.S., Norihan, A.H., Saleha, M.S.: IoT based facial recognition door access control home security system using raspberry pi. Int. J. Power Electron. Drive Syst. **11**(1), 417–424 (2020). https://doi.org/10.11591/ijpeds.v11.i1.pp417-424
28. Chao, W.-L.: Face recognition. GICE, National Taiwan University, Available online: https://www.orcam.com/en/myeye2/ accessed on 21 May 2022
29. Yang, S. Luo, P., Loy, C. C., Tang, T.: WIDER FACE: a face detection benchmark. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 5525–5533, (2016) https://doi.org/10.1109/CVPR.2016.596
30. Scherhag, U., Rathgeb, C., Merkle, J., Busch, C.: Deep face representations for differential morphing attack detection. IEEE Trans. Inf. Forensics Secur. **15**, 3625–3639 (2020). https://doi.org/10.1109/TIFS.2020.2994750
31. Khan, K., Khan, R.U., Ali, J., Uddin, I., Khan, S., Roh, B.H.: Race classification using deep learning. Comput. Mater. Contin. **68**(3), 3483–3498 (2021). https://doi.org/10.32604/cmc.2021.016535
32. N. I. o. S. a. T. NIST.: Face recognition technology (FERET). https://www.nist.gov/programs-projects/face-recognition-technology-feret. (2021)
33. Dwivedi, H., Android Instructor.: Comparing MobileNet models in TensorFlow. (2019) https://www.kdnuggets.com/2019/03/comparing-mobilenet-models-tensorflow.html accessed on 8 Mar 2022
34. Isuyama, V. K., Albertini, B. D. C.: Comparison of convolutional neural network models for mobile devices. Esc. Politecnica Univ. Sao Paulo (USP), Brazil, pp. 73–83, (2021) https://doi.org/10.5753/wperformance.2021.15724.
35 Wang, W., Li, Y., Zou, T., Wang, X., You, J., Luo, Y.: A novel image classification approach via dense-Mobilenet models. Mob. Inf. Syst. (2020). https://doi.org/10.1155/2020/7602384
36. Kadhim, T.A., Smaoui Zghal, N., Hariri, W., Ben Aissa, D.: Face recognition in multiple variations using deep learning and convolutional neural networks. In: 9th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. (SETIT'22), 2022., no. 1, (2022)
37. Khan, S., Rahmani, H., Shah, S.A.A., Bennamoun, M.: A guide to convolutional neural networks for computer vision. Synth. Lect. Comput. Vis. **8**(1), 1–207 (2018). https://doi.org/10.2200/s00822ed1v01y201712cov015
38. Liu, Y., Zhai, G., Zhao, D., Liu, X.: Frame rate and perceptual quality for HD video. In: Springer Int. Publ. Switz., vol. 9315, (2015) https://doi.org/10.1007/978-3-319-24078-7
39. Kiran, T.T.J.: Computer vision accuracy analysis with deep learning model using TensorFlow. Int. J. Innov. Res. Comput. Sci. Technol. **8**(4), 319–325 (2020). https://doi.org/10.2139/ssrn.3673214
40. Maruseac, M.: Support for 32 bits architecture. (2019) https://github.com/tensorflow/tensorflow/issues/32315
41. Adi, P. D. P., Kitagawa, A., Akita, J.: Finger robotic control use M5Stack board and MQTT protocol based. In: 7th Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2020—Proc., no. October, pp. 1–6, (2020) https://doi.org/10.1109/ICITACEE50144.2020.9239170
42. Ahmed, H. M., Rasheed, R. T.: A Raspberry Pi real-time identification system on face recognition. In: Proc. 2020 1st Inf. Technol. to Enhanc. E-Learning other Appl. Conf. IT-ELA 2020, pp. 89–93, (2020) https://doi.org/10.1109/IT-ELA50150.2020.9253107
43. Nikisins, O., Fuksis, R., Kadikis, A., Greitans, M.: Face recognition system on raspberry Pi. In: 2015 5th Int. Work. Comput. Sci. Eng. Inf. Process. Control Eng. WCSE 2015-IPCE, no. April, (2015) https://doi.org/10.18178/wcse.2015.04.054
44. Suchitra, Suja, P., Tripathi, S.: Real-time emotion recognition from facial images using Raspberry Pi II. In: 3rd Int. Conf. Signal Process. Integr. Networks, SPIN 2016, pp. 666–670, (2016) https://doi.org/10.1109/SPIN.2016.7566780
45. Lu, J., Fu, X., Zhang, T.: A smart system for face detection with spatial correlation improvement in IoT environment. In: 2017 IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust.

Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov. SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017, pp. 1–4, (2018) https://doi.org/10.1109/UIC-ATC.2017.8397550
46. Gsponer, D.: IoT: building a Raspberry Pi security system with facial recognition. *Haaga-Helia* (2018)
47. Kak, S. F., Mustafa, F. M.: Smart home management system based on face recognition index in real-time. In: 2019 Int. Conf. Adv. Sci. Eng. ICOASE 2019, pp. 40–45, (2019) https://doi.org/10.1109/ICOASE.2019.8723673
48. Munir, A., Kashif Ehsan, S., Mohsin Raza, S. M., Mudassir, M.: Face and speech recognition based smart home. In: 2019 Int. Conf. Eng. Emerg. Technol. ICEET 2019, pp. 1–5, (2019) https://doi.org/10.1109/CEET1.2019.8711849
49. Saputra, R., Surantha, N.: Smart and real-time door lock system for an elderly user based on face recognition. Bull. Electr. Eng. Inform. **10**(3), 1345–1355 (2021). https://doi.org/10.11591/eei.v10i3.2955
50. Orna, G., Benitez, D. S., Perez, N.: A low-cost embedded facial recognition system for door access control using deep learning. In: 2020 IEEE Andescon, Andescon 2020, pp. 0–5, (2020) https://doi.org/10.1109/ANDESCON50619.2020.9271984
51. Gunawan, T.S., Gani, M.H.H., Rahman, F.D.A., Kartiwi, M.: Development of face recognition on raspberry pi for security enhancement of smart home system. Indones. J. Electr. Eng. Inform. **5**(4), 317–325 (2017). https://doi.org/10.11591/ijeei.v5i4.361
52. Vamsi, T.K., Sai, K.C., Vijayalakshmi, M.: Face recognition based door unlocking system using Raspberry Pi Thulluri. Int. J. Adv. Res. Ideas Innov. Technol. **5**(2), 1320–1324 (2019)
53. Hasban, A. S. et al.: Face recognition for Student Attendance using Raspberry Pi. In: APACE 2019 - 2019 IEEE Asia–Pacific Conf. Appl. Electromagn. Proc., no. November, pp. 1–5, (2019) https://doi.org/10.1109/APACE47377.2019.9020758
54. Nagpal, G. S., Singh, G., Singh, J., Yadav, N.: Facial detection and recognition using OpenCV on Raspberry Pi Zero. In: Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018, pp. 945–950, (2018) https://doi.org/10.1109/ICACCCN.2018.8748389
55. Singh, S., Ramya, R., Sushma, V., Roshini, S., Pavithra, R.: Facial recognition using machine learning algorithms on Raspberry Pi. In: 4th Int. Conf. Electr. Electron. Commun. Comput. Technol. Optim. Tech. ICEECCOT 2019, pp. 197–202, (2019) https://doi.org/10.1109/ICEECCOT46775.2019.9114716
56 Nadafa, R.A., Hatturea, S.M., Bonala, V.M., Naikb, S.P.: Home security against human intrusion using Raspberry Pi. Procedia Comput. Sci. **167**, 1811–1820 (2020). https://doi.org/10.1016/j.procs.2020.03.200
57. Rok Novosel, B. M., Ziga Emersic, P. P., Struc V.: Face recognition with Raspberry Pi for IoT environments. In: *ERK Portorož*, no. September, pp. 477–480, (2017) https://lmi.fe.uni-lj.si/wp-content/uploads/2019/08/novoselface_recognition.pdf
58. Parthornratt, T., Burapanonte, N., Gunjarueg, W.: People identification and counting system using Raspberry pi. In: 2016 Int. Conf. Electron. Information, Commun., pp. 1–5, (2016)
59. Preetha, J., Manirathnam, M., Chaitanya, A., Raj, R.P.: Raspberry Pi based face recognition system. Int. J. Eng. Res. Technol. **8**(08), 1–4 (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.