Palestine Polytechnic University
College of Information Technology and Computer Engineering

Face Reminder System

Personal Friends Identification System using Face Recognition

Team:
Fadi Zagharneh
Rami Nawahda
Mohammad Karajeh

Supervisors:
Dr. Radwan Tahboub

Submitted to the College Of Information Technology And Computer Engineering in fulfillment of the requirements for the Bachelor degree in Computer System Engineering

Sep 2020

Acknowledgement

All praise and thanks are only for Allah, the one who, by his blessing and favor, perfected goodness, and good works are accomplished, the one who helped us get through this project, and gave us strength whenever we stumbled and faced hardships.

Most importantly, from the bottom of our hearts, we would like to extend the thankfulness to our wonderful parents and families, whom they shared our hopes, prayed for us, and supported us in good and bad.

For our wonderful supervisor Dr. Radwan Tahboub, all thanks for sharing his experience, ideas and sciences with us, he made us believe in ourselves and guided us through the whole process of the project by his excellent guidance, caring, and patience. Also, we would like to take this opportunity to thank Dr. Alaa Halawani Dr. Amal and Eng. Wael Takrouri for their kindness and help anytime we needed them.

Finally, deeply from our hearts, we would like to thank everyone who supports us to continue, believe and stand with us to reach here today, everyone with his name.

نبذة

من أكثر المواقف المحرجة التي يتعرض لها الشخص في حياته، والتي قد تسبب له مشاكل نفسية، هي عند لقائه بشخص قد التقى به سابقا ولا يستطيع تذكر اسمه، سواءً لضعف البصر أو النسيان بسبب مرض معين أو لفقد البصر، وعليه يهدف هذا المشروع الى عمل جهاز ذكي يعمل على تذكير المستخدم بإسم الشخص الذي أمامه، عن طريق كاميرا، وحاسوب صغير، وسماعة أذن، وزر للتحكم مثبتات على هيكل النظاره. حيث المطلوب من المستخدم عند اللقاء بشخص لا يتذكر اسمه الضغط على الزر، وبالتالي تقوم الكاميرا بأخذ صورة -صورة الوجه- لهذا الشخص وإرسالها الى الكمبيوتر ليقوم بمعالجتها واستخراج اسم الشخص المخزن سابقاً وارساله الى سماعة الأذن، ويستطيع المستخدم إدارة النظارة عن طريق تطبيق أندرويد، وهكذا قد تم حل مشكلة عدم التعرف على الشخص، التي من الممكن أن تحرج الشخص وتسبب له مشاكل نفسية.

Abstract

One of the most embarrassing situations that a person is exposed to in his life, which may cause him psychological problems, is when he meets a person he has met previously, and he cannot remember his name, whether due to poor eyesight, forgetfulness due to a specific disease, or vision loss, and therefore this project aims to make a smart device that works To remind the user of the name of the person in front of him, through a camera, a small computer, a headset, and a control button fixed on the frame of the glasses. Where the user is required when meeting a person whose name he does not remember to press the button, and therefore the camera takes a picture - the face image - of this person and sends it to the computer to process it and extract the name of the previously stored person and send it to the headset, the user can manage the glasses through Android application, and thus the problem of not recognizing the person has been solved, Which may embarrass the person and cause him psychological problems.

Table OF Contents

Acknowledgement **1**

نبذة **2**

Abstract **2**

List Of Images **4**

List Of Tables **4**

List Of Abbreviations **5**

1. Introduction **2**

1.1 Overview 2

1.2 Motivation 2

1.3 Scope 3

1.4 Objectives 3

1.5 Literature review 3

1.6 System Requirements 4

1.7 Constraints 4

2. Background **7**

2.1 Face Detection 7

2.1.1 Face detection using Haar Cascade Classifiers: 7

2.2 Face Recognition 9

2.2.1 K-Nearest Neighbor 10

2.3 System Backbone 12

2.3.1 Microcomputers 12

2.3.1.1 Classifications 12

2.3.1.2 Raspberry Pi Microcomputer 13

2.3.1.3 Software Background 14

2.3.2 Camera Module 15

2.3.3 Android smartphone and application 15

2.4 Hardware Options 15

2.4.1 Micro-Controllers 15

2.4.2 The Camera and Raspberry pi 16

2.5 Software Options 18

2.5.1 Operating System 18

2.6 Conclusion 18

3. System Design And Implementation **21**

3.1 Introduction 21

3.2 System Design 21

3.2.1 System Block Diagram 21
3.2.2 System Flowchart 22
3.3. Hardware Implementation 23
3.3.1 Installing Operating System On RasPi 23
3.3.2 Connect Raspberry Pi Camera 23
3.3.3 Connect Push Button 24
3.4 Software Implementation 24
3.4.1 Set Button GPIO pin (pin 3) 25
3.4.2 How Face Recognition Occurs 25
3.4.3 Android app development 26
3.4.4 How connection between RasPi and app done 30
3.5 Testing and Result 30
3.5.1 Testing 30
3.5.2 Result 31
3.6 Future Work 32
3.7 Conclusion 32

Appendix **33**

References **36**

List Of Images

- Figure 1. Haar Rectangles
- Figure 2. Biometric Identification and verification system
- Figure 3. Example of KNN classification
- Figure 4. Raspberry Pi parts
- Figure 5. Raspberry pi with camera module
- Figure 6. System Block Diagram
- Figure 7. The Flowchart of system
- Figure 8. Connection button with RasPi

List Of Tables

- Table 1. Classification Arduino and Raspberry Pi
- Table 2. Comparison between Raspberry Pi and Raspberry Pi Zero
- Table 3. Comparison between two options in OS

List Of Abbreviations

RasPi	Raspberry Pi
SD	Solid Driver
KNN	K-Nearest Neighbor
RAM	Random-Access Memory Technology
PIC	Peripheral Interface Controller
AVR	Advanced Virtual RISC
BSD	Berkeley Software Distribution
CSI	Clocked Serial Interface
OpenCV	Open Source Computer Vision
IDE	Integrated Development Environment
OS	Operating System
GUI	Graphical User Interface
GPIO	General Purpose Input/Output
GND	Ground
SFTP	Secure File Transfer Protocol
SSH	Secure Shell
SSID	Service Set Identifier
WiFi	Wireless Fidelity

Chapter One Introduction

1.1 Overviews

1.2 Motivation

1.3 Scope

1.4 Objectives

1.5 Literature review

1.6 System Requirements

1.7 Constraints

1. Introduction

This chapter contains an introduction to the project. Divided into overview, motivation, the scope of the project, objectives and literature review.

1.1 Overview

In this book, we will explain all the steps to build the smart glasses in detail, starting with the introduction chapter, background chapter, design and implementation chapter, appendix and references.

In the introduction chapter, a general description of the book, and we explain the motivations for choosing the idea of the project, the scope that the project will include, the objectives, literature review, requirements, the existing works, and at the end the constraints.

In the background chapter we explain how a person's face is detected, how a person's face recognizes all the system parts whether it is hardware or software, the design options for the project, and at the end we conclude what we use to build the project.

In the design and implementation chapter, we explain the theoretical part of the project building, including system block diagram, flowchart diagram, hardware implementation, software implementation, testing, conclusion, and at the end result and future work.
In the appendix and references, including questionnaire and references for the project.

1.2 Motivation

From the standpoint of completing the graduation project course to obtain an engineering degree in computer systems at palestine polytechnic university, and encouraging our supervisor Dr radwan tahboub with the idea of making smart glasses, for those who suffer from forgetting the names of his friends to remember their names when meeting them, and we also made a questionnaire to know the importance and need of this idea at the present time, and it became clear to us from the results of the questionnaire that this idea is important and we should complete it to help those who need it .

Note: To view the questionnaire, the questionnaire is added in the appendix section.

1.3 Scope

The project involves creating an intelligent system to remember previously stored person faces, with suitable computer, camera, headset, button, and battery.So system on computer detect face and recognize it ,then output face name on headset if exist else store new person.

System users are those who suffer from poor eyesight, forgetfulness due to a specific disease, or vision loss.

1.4 Objectives

This section is responsible for knowing the result that we need from the system.

- Build a smart system for face recognition with low cost and high accuracy.
- The system will know the face if it sees at the second time and tells the user name of this face.
- Near real-time response.
- Notify users when a new face is added.
- The system will be portable.
- Users control of the system by android application.
- Secure connection between application and glasses.

1.5 Literature review

As resulted from the questionnaire, the percentage of people who suffer from forgetting names has reached a percentage of 81% and the remainder may suffer from this problem in the future when the circle of relationships expands. So, there is a problem that needs to be solved and to solve this problem we need a system to help us to remember these names and maybe will help us to remember some notes about them all that just by one click.

It must be mentioned that there many attempts to build the same idea like our system contain cameras mounted on glasses which are called smart glasses but all those attempts had a

privacy barrier and our system will care about privacy issues and we will see that in chapter 4.

Additionally , the advantages smart glasses offer could be tremendous and should not be dismissed if we can avoid disadvantages.

Existing Work

A team in our university made something similar, their project was made to help blind people to find things around them, and their project depends on object detection. Also, another team made a project dependent on face recognition used in smart doors.

1.6 System Requirements

The system expected to fulfil the following requirements:

- Micro computer for run system and processing (face recognition software).
- Camera for input face image.
- Earphone to output names.
- Battery for powering the system.
- Android app for management.
- Button and wires.
- Glasses for carry system hardware.
- Use computer vision for face detection and recognition.
- Add, edit, and delete faces using the app.
- Image source camera and mobile gallery.

1.7 Constraints

This project is subject to the following constraints:

- The face of forgetting people must be saved in a database.
- Before starting to use the system, connections must be established between pi and app.
- Suitable illumination condition and suitable orientation for camera.
- Image must include one face in First enrollment for each person.

- The camera is facing the face.
- The person is away one meter from the camera.
- The system cannot be used in the night and dark areas.
- Must ask the person before taking his picture, to prevent privacy issues.

Chapter Two Background

2.1 Face Detection

2.2 Face Recognition

2.3 System Backbone

2.4 Scope

2.5 Conclusion

2. Background

This chapter covers all theoretical aspects. It provides a simple and clear description of the methods and techniques used in the system.

2.1 Face Detection

"Face detection just means that a system can identify that there is a human face present in an image or video". [1]
So, the goal of face detection is to determine if there are any face/s in the image or not, detect and locate faces in the image regardless of their position, orientation, scale, lighting, resolution and noises.
Face detection process is very hard on traditional computer code, because that need to store all faces in work environment, so that impossible, because that not practical way, so recently machine learning enters many technological fields, one of these fields computer vision that include image processing processes, such as face detection and face recognition.
so using machine learning makes face detection and recognition very easy.

There are many algorithms to detect faces in photos such as:

- Histogram of Oriented Gradients.
- Convolutional Neural Networks.
- Haar Cascade Classifiers.

But we are not going to clarify all of them and we will only explain the last one , because a Haar-Feature is manually determined while cnn determined by training, and HOG is puboular in human detection. .

2.1.1 Face detection using Haar Cascade Classifiers:

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object

Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. [2]

In haar classifier concerns about some common features that are found in common human faces like a dark eye region compared to upper-cheeks, a bright nose bridge region compared to the eyes and some specific location of eyes, mouth, nose… .

The characteristics are called Haar Features. The feature extraction process will look like in Figure 2**.**

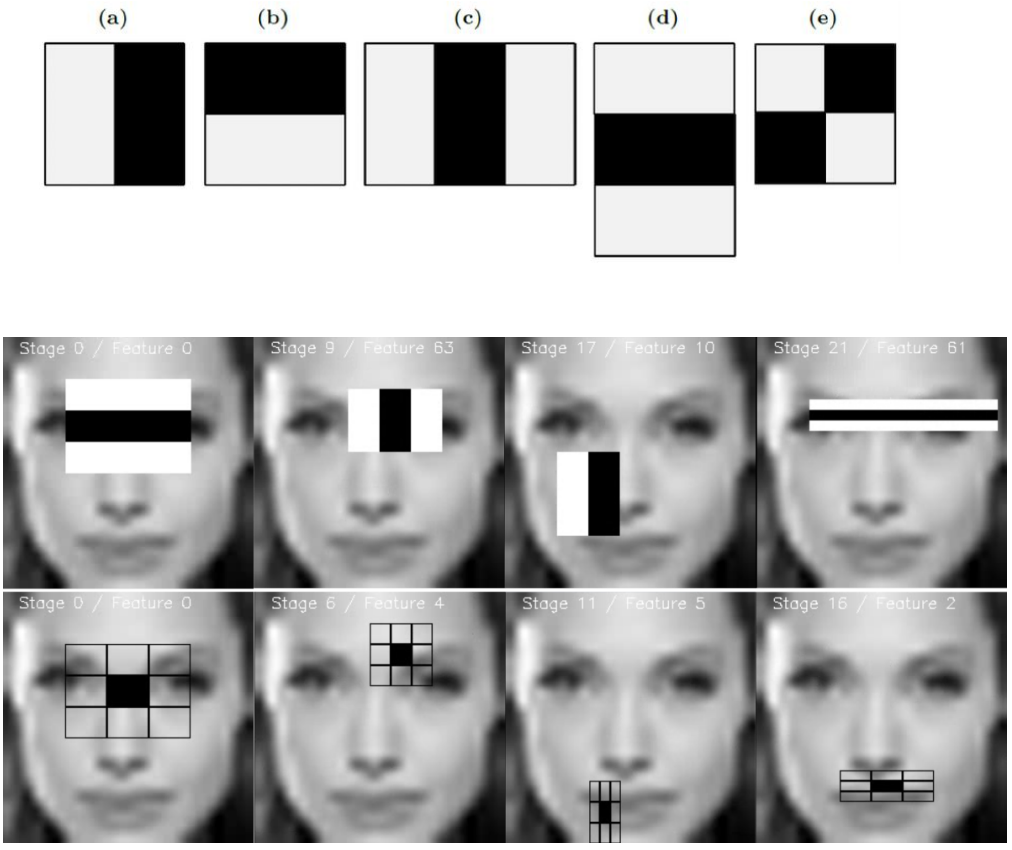


Figure 1. Haar Rectangles

Haar cascades in a simple way is take an image. Take each 24x24 window. Apply 6000 features to it.these feature check current window contains face or not, by inserting current window in multiple stages or classifier, each stage check window contains require features belong with current stage, if yes repeat until final stage, if no discard this window, because it not contains face.

2.2 Face Recognition

Face recognition has a significant appearance recently because of technological development, especially in computer vision. We can see face recognition in several fields or spaces, like airports, enterprise entrance, mobile authentication, monitoring locations, and many applications. So, what is face recognition, is an intelligent system that uses machine learning and complex calculation to extract what's known face print from the face image. The face recognition systems can operate basically in two modes:

- Verification or authentication of a facial image: it basically compares the input facial image with the facial image related to the user which is requiring the authentication. It is basically a 1x1 comparison.
- Identification or facial recognition: it basically compares the input facial image with all facial images from a dataset with the aim to find the user that matches that face. It is basically a 1xN comparison, we use this mode in the project.[3]

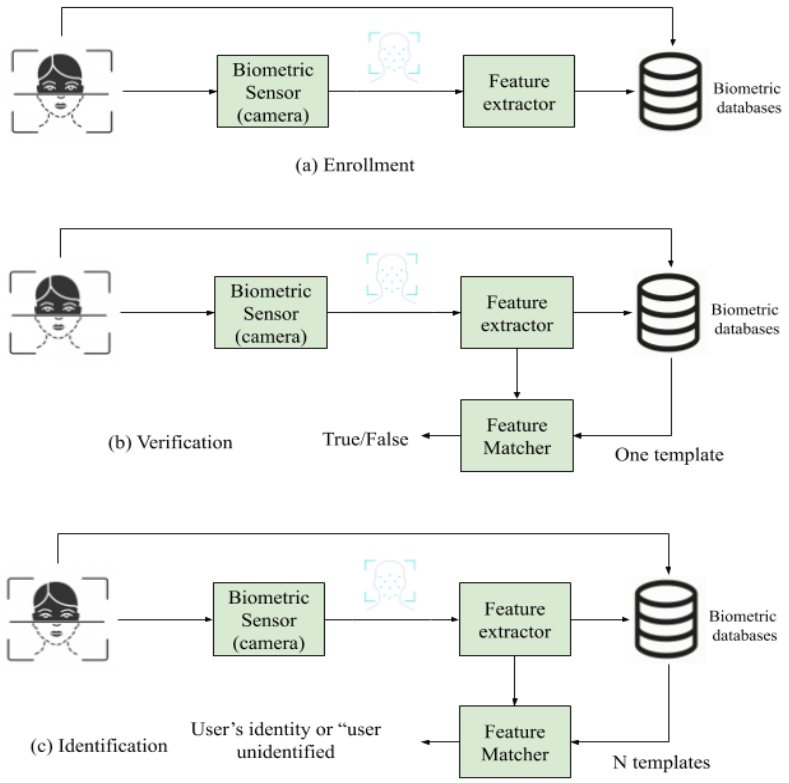


Figure 2. Biometric Identification and verification system

There are many algorithms used in face recognition, such as:

- Support vector machine.
- Convolution neural network
- K-Nearest Neighbor.

But we are not going to clarify all of them and we will only explain the last one, because is simpler than svm, and cnn need high memory.

2.2.1 K-Nearest Neighbor

KNN algorithm is among one of the simplest algorithms for regression and classification in supervised learning. KNN is non-parametric which means it does not make any assumptions but bases on the model structure generated from the data. KNN is called memory-based or lazy learning because the way it learns is just storing the representations of the training examples. An object is classified based on the majority votes of its neighbors (the training set). The new example object will be assigned to the class with its most similar k nearest neighbors. [4]

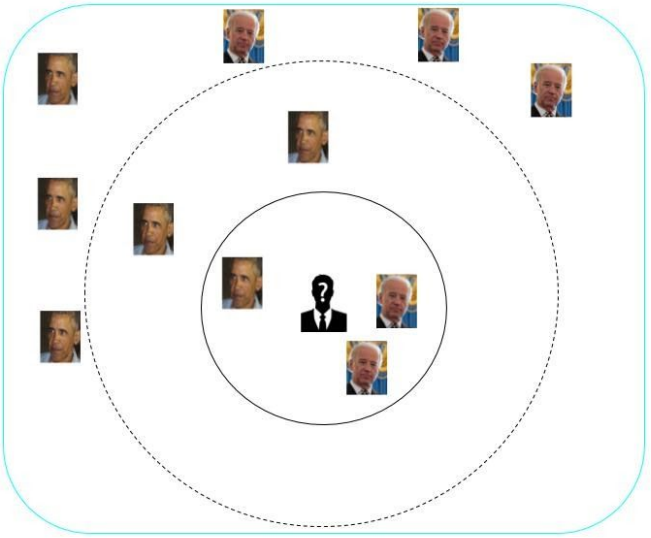


Figure 3. Example of KNN classification

In our project in the implementation section there are many steps from preparing the dataset and training the model to test our classifier.

The mathematical way to find the nearest face is the Euclidean distance metric which is by default in KNN. It is often chosen to determine the closeness between the data points in KNN. A distance is assigned between all pixels in a dataset. Distance is defined as the Euclidean distance between two pixels. The Euclidean distance is given by:

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + \ldots\ldots + (x_n - y_n)^2}$$

After finding euclidean distance between all known points and the unknown point, then set the name of the unknown point by taking the k nearest points name and the name will be the same name of more points of the same class.

2.3 System Backbone

This section will introduce information about the basic components of the system such as hardware components, which contains a microcomputer, and it's detailed, also the camera module which will be used. In addition android application information.

2.3.1 Microcomputers

"A microcomputer is a small, relatively inexpensive computer with a microprocessor as its central processing unit. It includes a microprocessor, memory, and input/output facilities". [5]

In the beginning, we could not determine the best microcomputer to use in the project, so we decided to make a comparison between the existing microcomputers in addition if we decided to implement the project using mobile.

2.3.1.1 Classifications

1) Microcontroller: it includes several types such as PIC, Arduino(contains AVR) etc. It has an embedded system consists of RAM, Input/output Pins and an oscillator that operates from 16-80 MHz

2) Microcomputer: Includes several types, such as Raspberry Pi, Beagle Bone. It consists of an internal clock, Input/output Pins and operates at a frequency from 700 MHz to 1GHz.

Module	Arduino	Raspberry Pi 3 Model B+
Price	30$	35$
Architecture	8Bit	32Bit
Clock Speed	16-20 MHz	700-1000 MHz
Random Access Memory	2 KB	512 MB
Flash Memory	32 KB	External SD Card(2 to 16G)
Multitasking	No	Yes
Memory	0.002MB	512MB
USB Port	One input only	Two peripheral OK
On-Board Network	None	10/100 wired Ethernet RJ45
Operating System	None	Linux distribution
Input voltage	7 to 12 V	5 V
Programming Language	Writing based (C/C++)	Python, C, Possibly Basic
Video	None	HDMI, NTSC or PAL

Table 1. Classification Arduino and Raspberry Pi

[6] [7]

2.3.1.2 Raspberry Pi Microcomputer

Raspberry pi

"The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games". [8] As we mentioned earlier in the classifications

section Raspberry pi has many preferences such as SD card for booting and long-term storage. Also, high speed to make image processing.

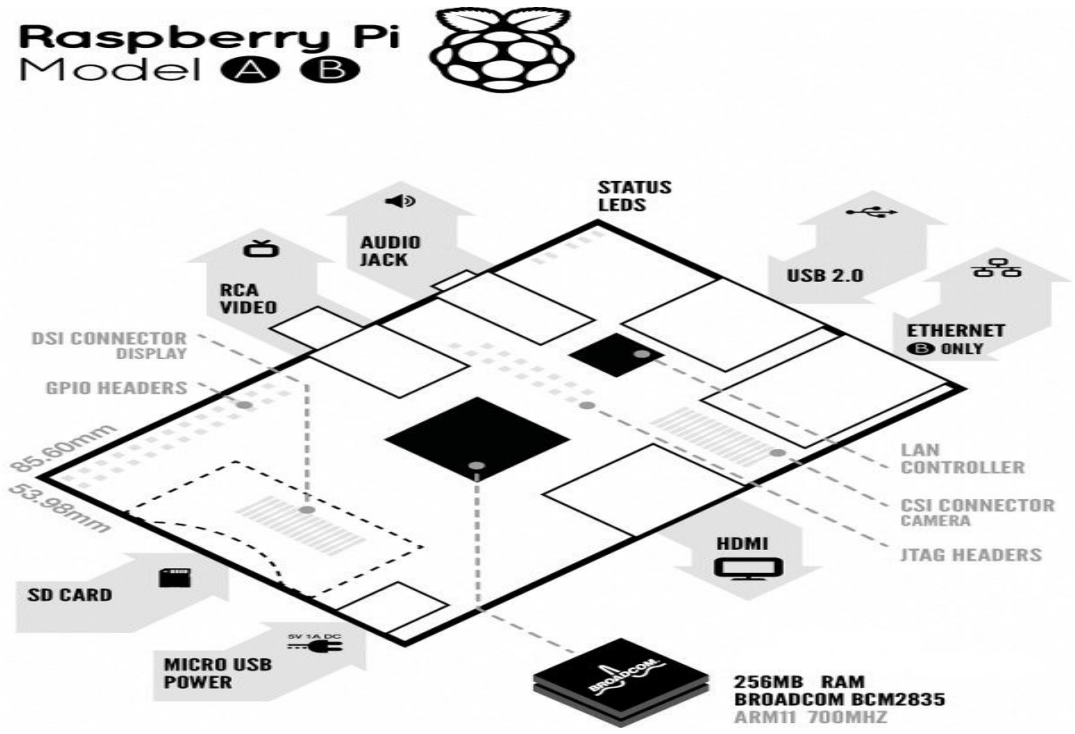


Figure 4. Raspberry Pi parts

2.3.1.3 Software Background

The main component of the system is the microcomputer. To handle this component we need an operating system to run and program it. Then we need a programming language to give the microcomputer the ability to process, send, and receive data. The programming language used within the raspberry pi called “Python”. Python is one of the most popular languages used in Raspberry Pi programming because it has two features: simplicity, power and easy learning.

Python’s Features:

1- Free and Open Source: In simple terms, anyone can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs.
2- Portable: Python has been ported (i.e. changed to make it work on) to many platforms. Python programs can work on any of these platforms without requiring any changes at all. Python can be used on Linux, Windows, FreeBSD, and Macintosh.

3- Extensive Libraries: The Python Standard Library is huge indeed. It can help to do various things involving regular expressions, documentation generation, ...etc.

2.3.2 Camera Module

The Pi camera module is a portable lightweight camera that supports Raspberry Pi. It communicates with Pi using the MIPI camera serial interface protocol. It is normally used in image processing, machine learning or in surveillance projects. It is commonly used in surveillance drones since the payload of camera is very less. Apart from these modules Pi can also use normal USB webcams that are used along with computer[9].

2.3.3 Android smartphone and application

In this project, we need an interface to handle a microcomputer such as enter data and review the people who have been stored, notified. Mobile has an Operating System called Android. Android worked with the OpenCV library so makes a mobile device an option to use. Android Application will be developed by Android studio IDE.

2.4 Hardware Options

The hardware components in our project have many options and types of the components that perform the desired tasks. In this section will compare between two of the most common categories of the major components as follows:

2.4.1 Micro-Controllers

The hardware components in our project have many options and types of the components that perform the desired tasks. In this section will compare between two of the most common categories of the major components as follows:

Model	Raspberry Pi 3 Model B+	Raspberry Pi Zero Wireless
RRP(USD)	$25.00	$10.00
SOC Type	Broadcom BCM2837B0	Broadcom BCM2835

Core Architecture	ARMv8-A (64/32-bit)	ARMv6Z (32-bit)
No. of Cores	4	1
GPU	Broadcom Video Core IV 1080p60	Broadcom Video Core IV 1080p30
CPU Clock	1.4 GHz Quad Core ARM Cortex-A53	1 GHz Single Core ARM1176JZF-S
RAM	512MB	512 MB
USB Ports / Min – Max (Power Input)	1 x USB / (Micro USB 5V @ 2.5A)	1 x Micro USB (OTG) / Micro USB 5V @ 2.5A
Wi-Fi	On Board Wi-Fi 802.11ac Dual Band 2.4GHz & 5GHz	On Board Wi-Fi 802.11n
Bluetooth	On Board Bluetooth 4.2/BLE	On Board Bluetooth 2.0/4.1
Video Output	HDMI 3.5mm Composite DSI (for LCD)	Mini HDMI Composite via PCB
Audio Output	I²S HDMI 3.5mm Composite	I²S Mini HDMI
Camera Input	15 Pin CSI	15 Pin CSI
No. of GPIO Pins	40	40 Header NOT Installed
GPIO Functions	17 x GPIO, UART, I²C, SPI, I²S, 1-Wire 3.3V/5V/GND ,EEPROM	17 x GPIO, UART, I²C, SPI, I²S, 1 Wire 3.3V/5V/GND, EEPROM
Memory	Micro SD	Micro SD
Length (mm) / Width (mm) / Weight (g)	65.0/ 56.0 / 29.0	65.0/ 30.0 /9.0

Table 2. Comparison between Raspberry Pi and Raspberry Pi Zero

Depending on the previous comparison we decided to choose the raspberry pi 3 b+, because its performance is very good to achieve face recognition.

2.4.2 The Camera and Raspberry pi

Initially, the camera is connected to the raspberry pi via a special serial port dedicated for the camera on the raspberry pi module. What follows is that this camera must be programmed and configured over the raspberry pi to run properly, in the configuration terminal many parameters can be set, such as resolution, frame rate, and delay between captured images.

The minimum image resolution allowed is 64 x 64 pixel, and the maximum camera resolution is 2592 x 1944 pixel with a maximum frame rate of 32, the transparency of the captured image and the brightness - which is an important factor when it comes to detect the faces in

the next step - are also configurable parameters. The camera will stream live video to the raspberry pi, in order for the detection algorithm to work with this stream.[10]

Figure 5 shows how the Raspberry pi camera module will be connected to the serial port on the raspberry pi microcomputer, this schematic illustrates the following:

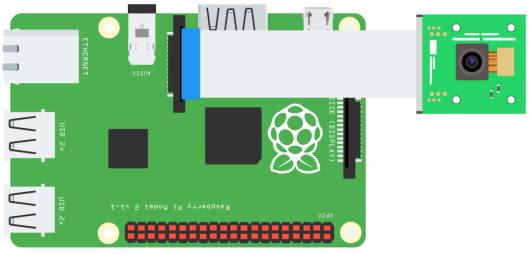


Figure 5. Raspberry pi with camera module

1) Programming the camera - after physically connecting it to the raspberry pi bored – to initialize.

2) A Python code will configure the camera to stream video.

3) Raspberry pi will read each frame in this video and apply the detection on each frame.

4) Whenever a face is present in the frame, the raspberry pi will detect and crop this face and save it on a folder on raspberry pi.

After the camera initialization and as the first frame arrives, the raspberry pi will start processing the incoming frames. Each frame will run through face detection algorithms, and faces, if they exist, will be detected and cropped. At the end of this stage, multiple captured faces will be stored on the SD card placed within the Raspberry pi; these faces were taken every time the camera detected them in the frames.

Mobile phones give us a more beautiful shape and its sensors if we choose Raspberry Pi Zero Wireless to take an image and send it to a mobile phone and it will process it.

Based on what was mentioned in the previous table, a choice will be made between three choices:

- Choice 1: Using Raspberry Pi Zero Wireless Alone.
- Choice 2: Using Raspberry Pi 3 Model A+ Alone.
- Choice 3: Using Raspberry Pi Zero Wireless with Mobile Phone.
- Choice 4: - Mobile Phone with Glasses has a camera.

Then, we have many choices depending on whether we need real time face recognition and good shape. So, after trying every option we will decide which is best.

2.5 Software Options

This section talks about software options that are compatible with the choices of hardware that are mentioned in the previous section, and what we chose and why.

2.5.1 Operating System

Depending on the physical components that will be used, the appropriate operating environment consisting of two options will be chosen as in Table 3:

OS	Raspbian	Android
Compatibility with hardware	Completely compatible with Raspberry Pi	Compatible with Android mobile phone and Raspberry Pi
Environment (language)	Python IDE (python) and more	Android Studio (Java)
Performance	High Performance	High Performance
Open source	Completely Open source	Completely open source
Add-ons	We can add cameras and other sensors.	We can use the advantage of sensors like GPS.
Supported Libraries	OpenCV (for Open Computer Vision)	OpenCV (for Open Computer Vision)

Table 3. Comparison between two options in OS

As mentioned in the table, we have more than one option as mentioned in the hardware design option. In our project, we will try two operating options and compare which gives better and faster results. There is no doubt that using the phone will give us more features than Raspberry Pi.

We can manage a microcontroller by itself with a touch screen module, or over SSH on a computer or mobile application to work as an interface.

2.6 Conclusion

After viewing the different types of face detection algorithms, face recognition algorithms and the system backbone, an initial decision is to work with the following configurations:

1) Face detection algorithm: Detection by a Haar Cascade with OpenCV.

2) Face recognition algorithm: KNN algorithm.

3) Raspberry Pi 3 model B+.

4) Raspberry Pi Camera Module 2.

5) Android smartphone.

Chapter Three
System Design And Implementation

3.1 Introduction

3.2 System Design

3.3 Hardware Implementation

3.4 Software Implementation

3.5 Testing and Result

3.6 Conclusion

3. System Design And Implementation

3.1 Introduction

This chapter explains the diagram and all steps to building the project, including building hardware and software components, to get the final product , which is smart glasses and its software.

3.2 System Design

This section contains a group of diagrams and flowcharts that illustrate the structure of the system and how it will work.

3.2.1 System Block Diagram

This subsection shows the general block diagrams of system hardware and software, which describe how the system components interfacing with each other as shown in Figure 6.

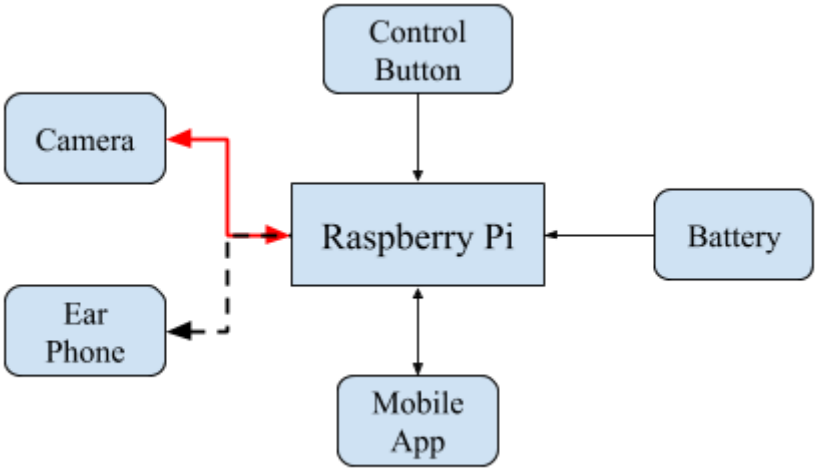


Figure 6. System Block Diagram

3.2.2 System Flowchart

This flowchart shows how the system works when it receives an image, and how it outputs the name of the face in the image.

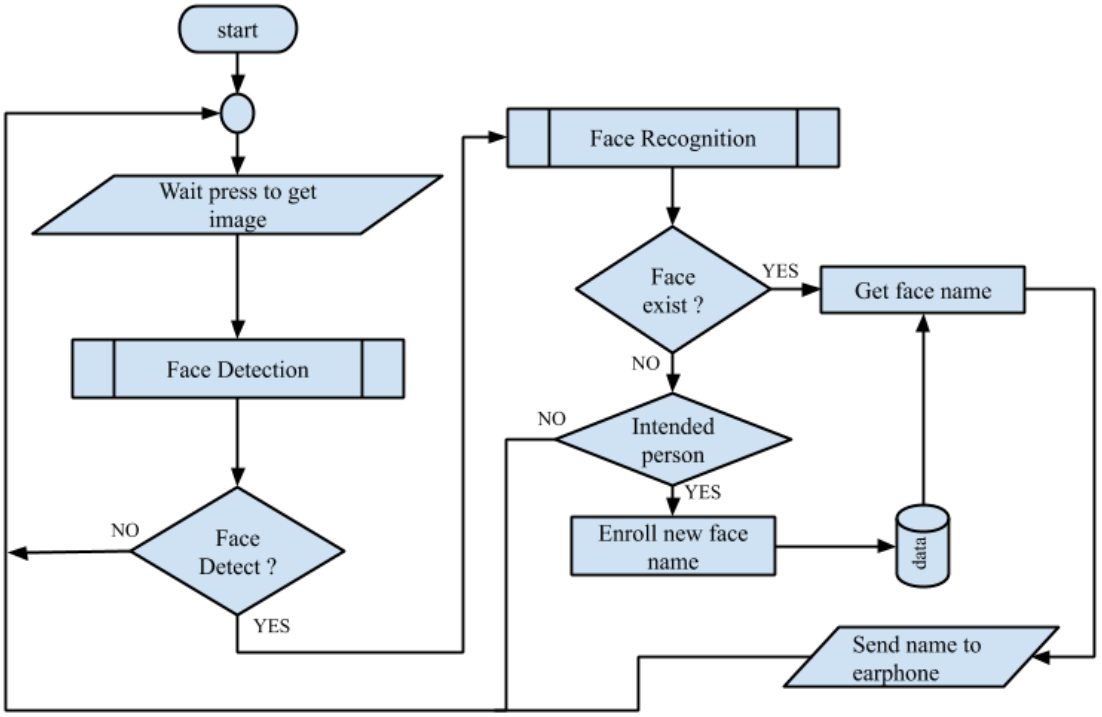


Figure 7. The Flowchart of system

3.3. Hardware Implementation

Build the hardware parts are described in this section.

3.3.1 Installing Operating System On RasPi

To install operating system on RasPi there are a lot of websites explain that, such sa raspberrypi.org, so in following simple steps you can do that:

1. Install the latest version of raspbian os image.
2. Write image on sd card by any image writer software.
3. Insert sd card into sd card slot in RasPi.
4. Connect the power to RasPi.
5. Wait until installing complet on RasPi.
6. You have done.

Note! Because we have many issues in installing opencv library, we install it in the same package of the raspbian, so you can do that by the same steps above but exchange raspbian package with package contains raspbian os and opencv library.

3.3.2 Connect Raspberry Pi Camera

Connect camera is so simple and straightforward, follow the steps below to do that:

1. Locate the Camera Module port.
2. Gently pull up on the edges of the port's plastic clip.
3. Insert the Camera Module ribbon cable; make sure the cable is the right way round.
4. Push the plastic clip back into place.
5. Start up your Raspberry Pi.
6. Go to the main menu and open the Raspberry Pi Configuration tool.
7. Select the Interfaces tab and ensure that the camera is enabled:
8. Reboot your Raspberry Pi.

The above steps are taken from raspberrypi.org website.

Connect Speaker:

Get any 3.5mm headphone and cut one speaker then connect it with the 3.5mm jack in RasPi board.

3.3.3 Connect Push Button

Add push button is very important, because this button is responsible for turning off/on RasPi and to take images to identify faces in this image.

To connect button successfully follow the below steps:

1. Connect button pins one with 5V pin and the other with resistor terminal.
2. Connect other resistor terminals with GND pins.
3. Connect the GPIO_IN in the **Figure 8** pin on RasPi GPIO to get button status.

The following image contains two types of connection, we use the input pull-low.

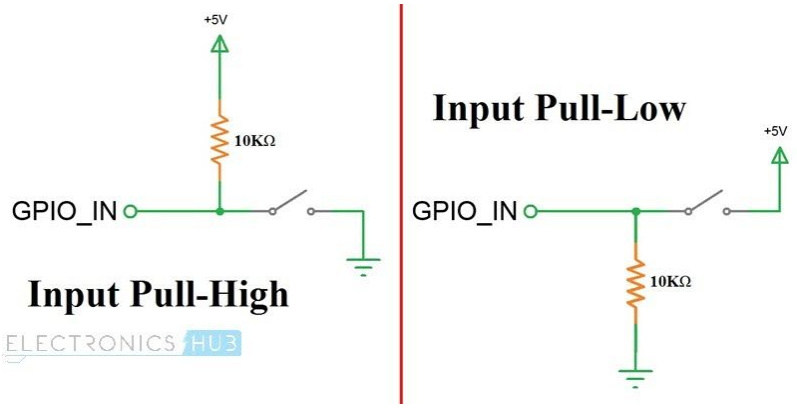


Figure 8. Connection button with RasPI

Connect Battery:

Power banks have more than 2A and mini usb cable then connect them in mini sub slote on RasPi.

3.4 Software Implementation

This section describes the software parts of the project, but Software parts divided into two parts are android application and python code setup on RasPi.So we get started with python code setup, then describe android app code.

3.4.1 Set Button GPIO pin (pin 3)

Connect button pin with pin 3 on RasPi, choosing this pin because of its functionality. This pin is set to handle push button click and match click with the following functions:

1. **Turn on**

   When RasPi is off state, and the user clicks on the push button, then RasPi will run, this happens because pi 3 has the property to turn on RasPi when it's off and connected with a power source.
2. **Turn off**

   if the click period is more than 4 seconds then RasPi will turn off automatically.
3. **Take picture**

   if the click period is less than 4 seconds then RasPi will take a picture.

Note! In point 2 and 3, click on pin 3 is set as interrupt, if interrupt occurs then depending on click time action occurs (turn off or take picture).

3.4.2 How Face Recognition Occurs

Hint: import necessary packages (face recognition, opencv, and pyttsx3).
we need the following folders/files:

1. Known faces folder.
2. Unknown face folder.
3. Face recognition file.
4. Train model file, this initializes the face model file after the first run if known faces folder not empty.
5. Faces model, depend on it to find matches between faces.
6. Face model, to detect faces in image.
7. And other folders and files necessary for the project.

If push button is clicked less than 4 seconds then the following steps will occur:

a. get a picture and resize it to 400 x 400 pixels.

b. Detecte faces by haar Cascade Classifier as the following:

```
Faces = face_recognition.face_landmark(photo_url)
```

c. If face exists then:

   i. If face recognized successfully output face name to speaker like this:

   ```
   text_to_speech("this is" + face_name + "face")
   ```

   Text to speech function contains code to convert text to voice by pyttsx 3 library.

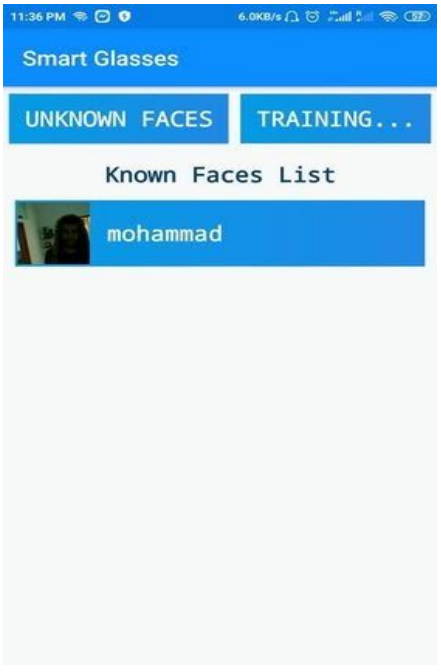
   ii. If not recognized successfully, then take five pictures for the current face and save them in an unknown faces folder.

d. If face not exists then the program will return to the main state and output "no face exists" by text to speech function.

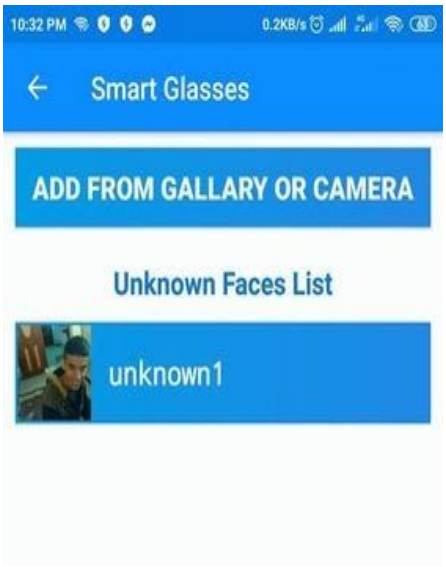
3.4.3 Android app development

This app using to simplify actions on raspi, so app contains the following pages:

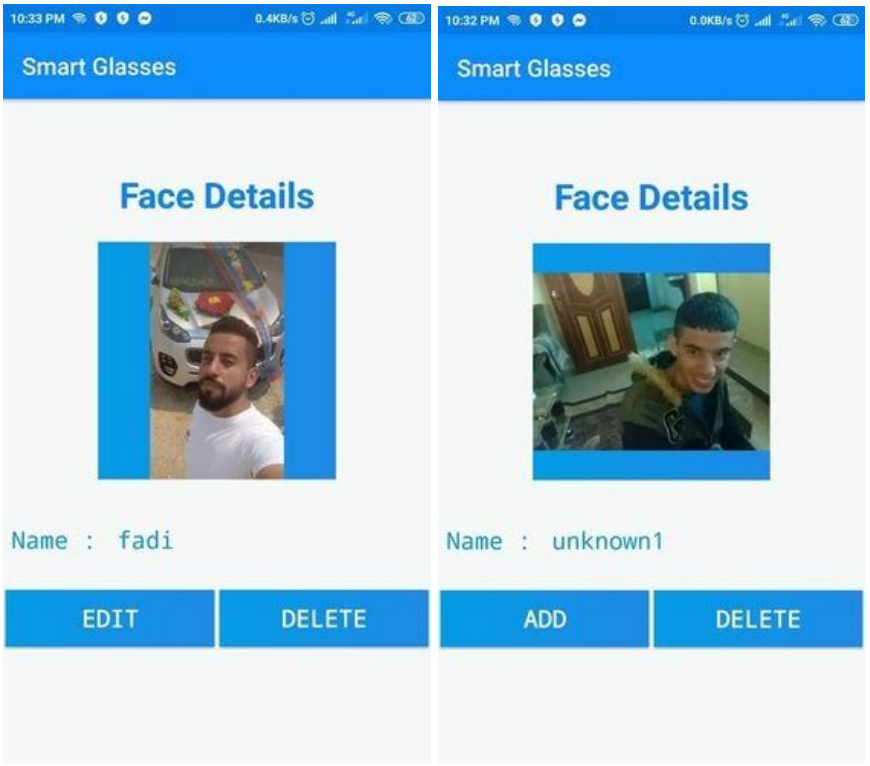
1. Main page contains a button to train faces model classifier, two radio buttons to set face recognition type, button to navigate to unknown faces page, and list to view known faces.



2. Unknown faces page, contains button to pick image from gallery, back arrow to back to main page, and list to view unknown faces.



3. Face details page, contains face image, button to add/edit face, face name,and button to delete face.



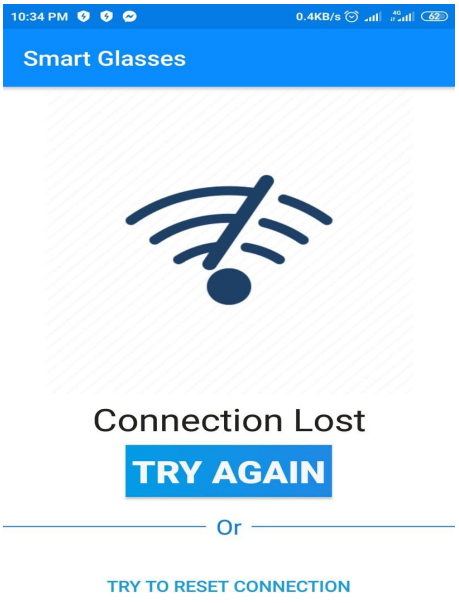
When click Edit view dialog like this:

When add is clicked view dialog like this:

.

4. Set connection page, contains text views, text fields, and button to connect.

5. Lost connection page, contains image to describe connection lost, button to reconnect, and button to set connection.



Connection between app and pi over SSH with execute shell and SFTP channel.

3.4.4 How connection between RasPi and app done

Create an execute channel to run specific files in RasPi.

```
Channel channel = session.openChannel("exec");
((ChannelExec) channel).setCommand("python3 train.py'");
```

Create a channel to get and send image files, edit folders name, move folders, and delete folders, this channel is called secure file transfer protocol (SFTP) channel.

```
Channel channel = session.openChannel("sftp");
ChannelSftp sftp = ((ChannelSftp)channel);
```

The Jsch library is responsible for all that.

3.5 Testing and Result

3.5.1 Testing

This section will go through many tests that were accomplished over the system.These tests consist of four main parts, which are:

1. Testing detection-processing time.
2. Testing recognition-processing time.

Note: we used five persons (samples) and each person had five face images.

Detection Cases:

Case	Time (second)
Image without faces	3.712
Image with face	0.895

Recognition Cases:

Image with known face	6.15
Image with unknown face	7

Inserting New Face:

Case	Time
Image with unknown face and add it.	13.5

Training Test:

Class	Time
3 Sample	126.42
5 Sample	188.89
10 Sample	408.23

Threshold Value:

Sample = 5

Threshold	Time	Accuracy
0.8	Short	50%
0.6	Middle	80%
0.4	High	85%
0.2	Very High	90%

Note: we chose 0.45 because it has practical time and 87% accuracy.

3.5.2 Result

Class	Value
Threshold	0.45
Number of person for each person	5
K value for KNN	5

3.6 Future Work

After finishing the required objectives in this project, the final suggestions and recommendations for this project are:

1. To improve the system efficiency, use a higher version of the RasPi model like the 4 version or ASUS Tinker Board which contains an integrated Graphics Processor.
2. A lighting sensor can be added to manipulate the detected faces equalization by measuring the current light intensity and upon that value a specific equalization is done.
3. High camera resolution can be more efficient also and if it is tiny this is better and .
4. Use cloud technology like store all databases on Firebase Realtime Database or use it to connect between RasPi and mobile. Also, we can reach a high efficiency result by using TensorFlow and Keras.
5. Add feature like read text, language translation, and location detection.

3.7 Conclusion

At the end of this project we build smart glasses and its android application. Smart glasses when clicking on the button get an image, detect the face, recognize the face name, and output name by the headset. Android application insert/delete new unknown person, edit/delete known person, and alert user with notification when new person added. Finally this system can help those people that suffer from low/loss vision or forgetfulness.

Appendix

نسيان اسماء الأشخاص

تم عمل هذا الاستبيان لمساعدتنا لاتمام مشروع تخرجنا، لمعرفة نسبة الأشخاص الذين تواجههم هذه المشكلة في حياتهم اليومية.

*Required

* هل سبق والتقيت بأشخاص للمره الثانية ولم تتذكر اسمهم؟

- تذكرت شكله ولم أتذكر اسمه
- تذكرت شكله و اسمه
- لم أتذكر شكله ولم أتذكر اسمه

* هل تعتقد انه يمكن للتكنولجيا حل مشكلة نسيان اسماء الأشخاص؟

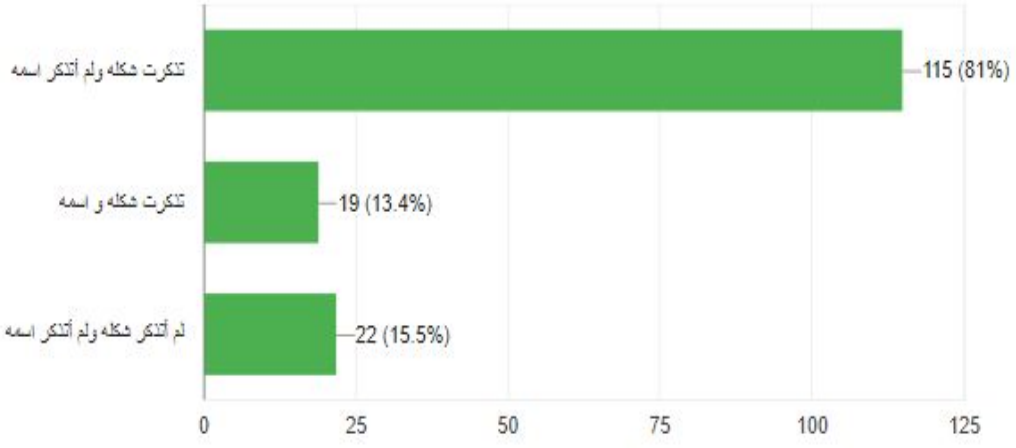
- نعم
- لا
- ربما

ما هذه الطريقة بنظرك؟

Your answer

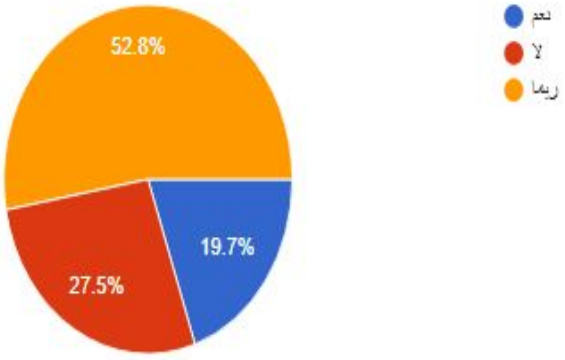
هل سبق والتقيت بأشخاص للمرة الثانية ولم تتذكر اسمهم؟

142 responses



هل تعتقد انه يمكن للتكنولجيا حل مشكلة نسيان اسماء الأشخاص؟

142 responses



ما هذه الطريقة بنظرك؟

References

[1] "Face Detection Vs. Face Recognition: What's the Difference?." 28 May. 2017, https://www.facefirst.com/blog/face-detection-vs-face-recognition/.

[2] "Face Detection using Haar Cascades — OpenCV-Python ...." http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html.

[3] "Face Recognition: Understanding LBPH Algorithm | by Kelvin ...." 10 Nov. 2017, https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b. Accessed 12 Sep. 2020.

[4] "K-Nearest Neighbor in Machine Learning - KnowledgeHut." 23 Sep. 2019, https://www.knowledgehut.com/blog/data-science/knn-for-machine-learning. Accessed 29 May. 2020.

[5] "What does Microcomputer mean? - Definitions.net." https://www.definitions.net/definition/Microcomputer. Accessed 28 May. 2020.

[6] "comparison - Adafruit Blog." 18 Jun. 2012, https://blog.adafruit.com/2012/06/18/ask-an-educator-whats-the-difference-between-arduino-raspberry-pi-beagleboard-etc/comparison/. Accessed 28 May. 2020.

[7] "Arduino Vs. Raspberry Pi: Which Is The Right DIY Platform For ...." 7 May. 2014, https://readwrite.com/2014/05/07/arduino-vs-raspberry-pi-projects-diy-platform/. Accessed 28 May. 2020.

[8] "What is a Raspberry Pi?." https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/. Accessed 28 May. 2020.

[9]"https://components101.com/misc/pi-camera-module#:~:text=The%20Pi%20camera%20module%20is,learning%20or%20in%20surveillance%20projects".

[10] "Buy a Camera Module V2 – Raspberry Pi." https://www.raspberrypi.org/products/camera-module-v2/. Accessed 29 May. 2020.