



Camera Security System with Facial Recognition and Motion Detection

Authors:

Cristina Nita: G00332787

Stoyan Rizov: G00313180

B.Sc. (Hons) in Software Development (Level 8) - Galway-Mayo Institute of Technology (GA_KSOAG_H08),
Department of Computer Science & Applied Physics
April 2019

Supervisor:

Dr. Martin Hynes

Declaration

I hereby declare that both authors have contributed to writing this documentation and that it is entirely our own work. The material we have presented to you has not been taken from the work of others, except work that has been clearly referenced within this documentation.

Signed: _____

Date: 26 April 2019

Table of Contents

| | |
|---|-----------|
| DECLARATION..... | 2 |
| TABLE OF ILLUSTRATIONS..... | 5 |
| TABLE OF TABLES..... | 5 |
| TERMINOLOGY | 6 |
| ACKNOWLEDGEMENTS..... | 7 |
| ABSTRACT..... | 7 |
| 1. INTRODUCTION | 8 |
| 1.1 PROJECT DESCRIPTION | 8 |
| 1.2 PROJECT AIM..... | 8 |
| 1.3 PROJECT OBJECTIVES | 9 |
| 1.3.1 <i>Generating the Database</i> | 9 |
| 1.3.2 <i>Facial Recognition Application Development</i> | 9 |
| 1.3.3 <i>Mobile Application Development</i> | 10 |
| 1.3.4 <i>Implementing Communication between the two Applications</i> | 10 |
| 1.3.5 <i>Testing Facial Recognition Application on Raspberry Pi</i> | 11 |
| 1.4 PROJECT PLANNING..... | 11 |
| 2. METHODOLOGY | 13 |
| 2.1 RESEARCH..... | 13 |
| 2.1.1 <i>Project Commencement</i> | 14 |
| 2.2 ACTUAL DEVELOPMENT..... | 14 |
| 2.2.1 <i>Agile Methodology</i> | 14 |
| 2.2.2 <i>Development</i> | 15 |
| 2.2.3 <i>Testing</i> | 15 |
| 2.2.4 <i>Supervisor Meetings</i> | 16 |
| 3. TECHNOLOGY REVIEW | 17 |
| 3.1 INTRODUCTION..... | 17 |
| 3.2 FACIAL RECOGNITION SYSTEM DEVELOPED | 18 |
| 3.2.1 <i>Python: Software Programming Language for the Server</i> | 18 |
| 3.2.2 <i>Spyder: Integrated Development Environment (IDE)</i> | 20 |
| 3.2.3 <i>Libraries</i> | 20 |
| 3.2.3.1 <i>Haar Classifier: Face Recognition Library</i> | 20 |
| 3.2.3.2 <i>OpenCV: Motion Detection Library</i> | 21 |
| 3.2.3.3 <i>Tkinter: GUI Library</i> | 22 |
| 3.2.4 <i>Hardware used in Facial Recognition</i> | 22 |
| 3.2.4.1 <i>Raspberry Pi</i> | 23 |
| 3.2.4.2 <i>Camera</i> | 24 |
| 3.3 ANDROID APPLICATION DEVELOPMENT | 24 |
| 3.3.1 <i>Java: Software Programming Language for the Client</i> | 25 |

| | | |
|---------|---|-----------|
| 3.3.2 | <i>Android Studio: Integrated Development Environment (IDE)</i> | 25 |
| 3.4 | FIREBASE: CLOUD SERVICE PROVIDER | 28 |
| 3.4.1 | <i>Firestore Cloud Service Uses</i> | 29 |
| 3.4.1.1 | <i>Image and Video storage.....</i> | 29 |
| 3.4.1.2 | <i>Authentication.....</i> | 30 |
| 3.4.1.3 | <i>Pusher Beams: Sending Notifications from the Server to the User ..</i> | 30 |
| 3.5 | FRAMEWORKS USED | 31 |
| 3.5.1 | <i>Windows 10</i> | 31 |
| 3.5.2 | <i>Raspbian.....</i> | 32 |
| 3.5.3 | <i>Android.....</i> | 32 |
| 4. | SYSTEM DESIGN..... | 33 |
| 4.1 | ARCHITECTURE | 33 |
| 4.2 | DATABASE DESIGN..... | 34 |
| 4.3 | SERVER DESIGN AND FUNCTIONALITY | 36 |
| 4.4 | CLIENT DESIGN AND FUNCTIONALITY | 36 |
| 3.5.4 | <i>Registration</i> | 36 |
| 3.5.5 | <i>Login.....</i> | 37 |
| 3.5.6 | <i>Home</i> | 37 |
| 5. | SYSTEM EVALUATION | 39 |
| 5.1 | ROBUSTNESS AND EFFICIENCY | 40 |
| 5.2 | SCALABILITY | 41 |
| 5.3 | SECURITY AND VALIDATION | 42 |
| 6. | CONCLUSION | 43 |
| 7. | APPENDICES..... | 44 |
| 7.1 | GITHUB LINK..... | 44 |
| 8. | REFERENCES..... | 45 |

Table of Illustrations

| | |
|---------------------|----|
| FIGURE 1 | 15 |
| FIGURE 2 | 17 |
| FIGURE 3 [22] | 21 |
| FIGURE 4 [46] | 28 |
| FIGURE 5 [53] | 29 |
| FIGURE 6 [57] | 31 |
| FIGURE 7 | 34 |
| FIGURE 8 | 34 |
| FIGURE 9 | 35 |
| FIGURE 10 | 35 |
| FIGURE 11 | 37 |
| FIGURE 12 | 37 |
| FIGURE 13 | 38 |
| FIGURE 14 | 38 |
| FIGURE 15 | 38 |
| FIGURE 16 | 38 |
| FIGURE 17 | 39 |
| FIGURE 19 | 39 |
| FIGURE 18 | 39 |
| FIGURE 20 | 39 |
| FIGURE 21 | 41 |

Table of Tables

| | |
|----------------------------|----|
| TABLE 1: MILESTONES | 12 |
| TABLE 2: GANTT CHART | 12 |

Terminology

| | |
|---------------------------------------|--|
| Active devices | Defined as devices receiving at least one push notification per month. |
| Central Processing Unit (CPU) | The unit performing the processing inside a computer. |
| Framework | A layered structure indicating what kind of programs can or should be built and how they would interrelate. |
| Gaussian Mixture Model | A set of data points categorized into clusters based on similarity. Also known as Clustering. |
| Graphical User Interface (GUI) | An interface that uses visual indicators to interact with electronic devices such as computers and smartphones. |
| Library | A collection of non-volatile resources used by computer programs, often for software development |
| Linux Distribution | An operating system made from a Linux kernel based software collection. |
| Operating System (OS) | The low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals. |
| Platform | The environment in which a piece of software is executed. |
| Software Development Kit (SDK) | A set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platforms. |
| User Interface (UI) | Is the design of user interfaces for machines and software. Also known as user interface engineering. |

Acknowledgements

I would like to thank my supervisor, Dr. Martin Hynes for his support and for guiding us throughout this project. I would also like to thank the head of the year, Dr. John Healy for his commitment to instruct us about the importance of learning new technologies as part of this final year project.

Abstract

Home security is a very important tool to own in society nowadays as the usual cause of an entry or burglary can be the low levels of protection, therefore we decided to create a camera security system that includes facial recognition, motion detection and notifications sent to the user's mobile devices. Using mobile phones to check the household's breach in security increases portability in any location in which internet is available.

The project is also integrated with an emergency button capable of dialing an emergency number such as the police with a click of a button making it easy to react at urgent situations. The camera is capable of recording the footage of friends or un-identified people, as per user's choice, and storing it on the cloud making it easier to be distributed within the users that are registered within the household and also easing the access of checking on the house.

1. Introduction

1.1 Project Description

According to research undertaken by Phone Watch, “a burglary takes place in Ireland every 31 minutes” [1] . County Dublin is one of the most affected counties with a rate of 45% out of all the burglaries.

Due to the successful invention of the Alarm Systems that people have started to install in their households, such crimes have decreased significantly in most locations. Unfortunately, not the same can be said about County Limerick which is said to have experienced a 14% increase while County Clare’s increase was higher by 5%. Even with the fortunate drop in the amount of theft cases in the past ten years – since 2009 – it still “remains the most common category of offence” [2] accounting for 67,127 thefts. This is the highest number of Recorded Crime Offences [3]. To prevent such events from happening, we came up a project proposal involving using a camera to watch the front door of the house. The camera then detects the people that show up at the door and can also take videos of their actions.

Image Recognition involves recognizing people, animals, objects using machine learning concepts as well as different algorithms. The technology is linked to the camera and the raspberry pi and account for the main functionality of this project.

To be able to train the data, the application controlling the its Image Recognition requires pictures of the people that it is supposed to identify. Those images are uploaded to Firebase via a mobile phone application. The server then downloads those images and uses them to train the data and classify the people it sees as visitors or intruders.

1.2 Project Aim

The aim of this project is to develop a security system that uses facial recognition to identify the people approaching the user’s place of residence and to distinguish between those known by the user and strangers. The ability to differentiate visitors and intruders can be very useful in bringing about awareness to the customer in case of a breach in security, if such arises.

1.3 Project Objectives

The main objectives of this project are the following:

1. Generating a database that retains visual data – images – to train the image recognition program
2. Setting a facial recognition server to differentiate between strangers and the user's acquaintances
3. Developing a mobile phone application to allow the server to communicate with the client and vice versa
4. Implementing the communication between the mobile application and the server to make sure the customer is receiving notifications
5. Moving the completed application from the Personal Computer to the Raspberry Pi and testing its functionality to make sure it works as required

1.3.1 Generating the Database

The database we decided to use to store the pictures for this project is called Firebase and it is one of the top 50 most popular databases [4] used by developers. The advantage is that this database not only works with both Python and Java – in Android Studio – languages, but it also works on Linux which is an Operating System we wish to be working with as part of our learning curve.

This cloud service provider was used for multiple purposes including storing videos taken of the visitors and images required for training, allowing communication between the client and the server as well as storing the user's Authentication information.

1.3.2 Facial Recognition Application Development

The core of this project involves building an application to sync data from the cloud as well as video recordings from the camera of people identified as unknown. For this we require setting up a facial recognition system for the camera to determine the user's acquaintances and to video capture the faces of those not matching the recorded data.

There are many ways to set up this technology, the most used one is the Convolutional Neural Network [5]. However, the technique we used was not Convolutional Neural Network but instead we used OpenCV which

was started at Intel in 1999 by Gary Bradsky and released in the year 2000. OpenCV stands for Open Source Computer Vision Library, originally the library is written in C/C++, but it also has interfaces for python and java. In our case, we used python. The library itself is very quick and has excellent performance because it was originally written in C/C++, and it is also easy to deploy. This makes it an excellent choice for computationally intensive programs [6]

1.3.3 Mobile Application Development

The development of this technology was necessary to allow the users to upload pictures of their family, friends and any other visitors to the cloud. The importance of using a means of technology that is always in the customer's reach was the reason why we chose a mobile application.

Most people use their phones to take pictures of others as supposed to a digital camera. They are a lot smaller than most cameras, not as heavy and most of them can fit in our pockets. This makes them a lot more comfortable and convenient to use; which is mainly why we decided on a mobile application as supposed to a computer.

This app is also used to notify the users if someone they don't know is seen on the camera. As the main purpose of our project is to prevent burglaries and most importantly, household thefts; this was a very significant part of it.

The user can download videos taken by the camera into their phone memory and have it as visual proof against a possible culprit. If it is clearly visible in the video that a burglary is occurring and the user is not at home to intervene, then he or she can dial 911 directly from the application.

1.3.4 Implementing Communication between the two Applications

An important yet challenging part would be to create a means of communication between the server and the client. This is necessary for sending notifications to the device to let the user know that there is someone at the door. If it is someone the user knows, the name of that visitor is also sent via the notification.

As mentioned, the cloud service provider is Firebase. It is very efficient in allowing information delivery between the client and the server. This was done using an API built especially for it called Pusher Beams. Pusher is the category leader [7] for building collaboration features and Beams [8] is one of the products it provides. This API was used to send notifications from the server to the android app.

We found out that it only works with clients and servers written in specific, but you can also program alternative languages to suit your needs [9]. Fortunately, we did not need to do that as Java – being the language used for our Android App – was one of the languages that supported the client side and Python – which is what was used for the development of our Facial Recognition program – was the server side [9].

1.3.5 Testing Facial Recognition Application on Raspberry Pi

Because there is no way of connecting the application to the outside door with a laptop, we have decided the Raspberry Pi was a significant requirement. The only way of connecting the camera to the server using a computer, to be able to place it outside the home, was to use this technology. This is a credit-card sized computer [10] and it is compared to the size of the personal computers which is very convenient for the intended use.

However, Linux [11] seemed a better option as most corporate environments involved in Software Development are using it. This Operating System is free compared to Windows which is not and that is a huge benefit for a company.

1.4 Project Planning

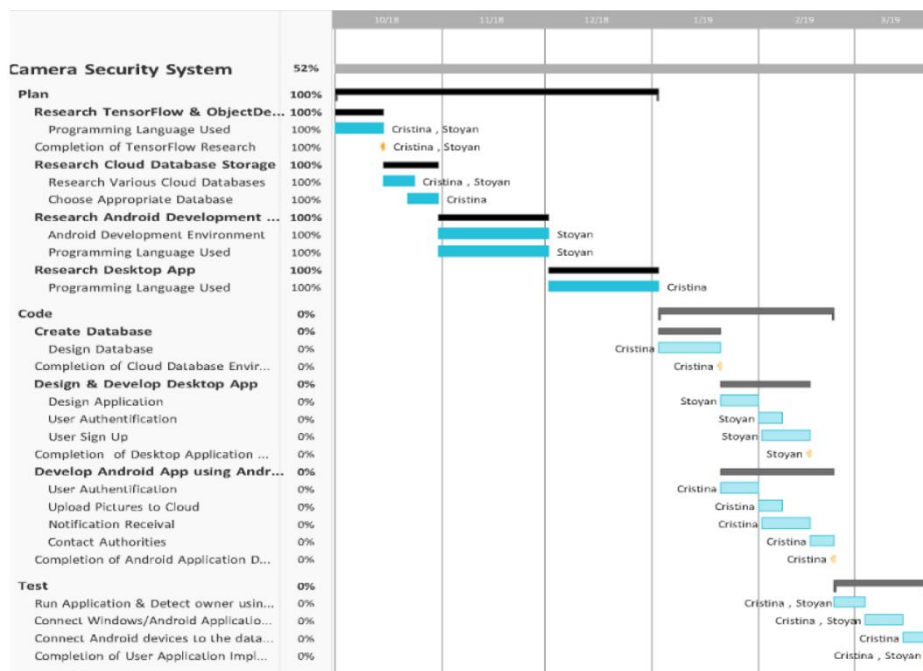
Table 1: Milestones, as the name suggests, the table shows the milestones of the project. These determine and monitor the progress of the project.

The most important tasks to be completed for this project as well as the time milestones and their percentage of completion Table 2: Gantt Chart.

Table 1: Milestones

| Milestone in Chronological Order | Tasks to be completed |
|----------------------------------|--|
| 1 | Conclude Extensive Research in relation to Facial Recognition, Cloud Service Providers, Native Applications and Additional Hardware to be used |
| 2 | Create the cloud database environment |
| 3 | Complete the development of a Desktop Application |
| 4 | Achieve a working Android Application to communicate with the user |
| 5 | Accomplish User Interface implementation for both the server (Raspberry Pi) and the client (Android Application) |
| Aim | A working surveillance system that uses Facial Recognition and a Camera and lets the user know who has visited the home |

Table 2: Gantt Chart



2. Methodology

2.1 Research

In this section, the project planning will be discussed as it is necessary to have a plan before starting in order to avoid running into problems and errors in the future. Especially when it comes to working in a team, planning is crucial before any source code is to be written, as it could lead to confusions and poor programming layouts and practices.

The first step after the project requirements was released in September was to come up with a good project idea that would be both creative and meet every requirement by the document. We also wanted to make sure the project would involve both of us working with something we haven't before in order to improve our programming background. As soon as the idea was decided, we had to choose the programming languages that were to be used, after some extensive research, to make sure those languages would best suit our idea's core tasks. After the decision was made regarding the idea and the technologies, we discussed it with the supervisor and decided to begin the project.

After coming up with the idea we had to brainstorm how it was going to be put into coding logic. For example, here are few questions that we had to ask ourselves before starting to code: How will videos be stored? Where will they be stored? Will the server run on a Raspberry Pi or laptop? So that made us think about the headings I will discuss below.

Architecture: What type of architecture are we going to apply to our project.

Technologies: The different technologies that are available to us, what is the best technology to use to achieve the best result. The frameworks that will be used and the different environments that will be needed.

Project Management Tools: The management tools that we will need for communication between team members, where will the code be centralized.

Testing: The testing techniques that will be used for our project into creating a robust programming style.

Deployment: The options that we have on deploying the solution for the project.

2.1.1 Project Commencement

It was then time to begin our project, so the meeting with the supervisor was arranged, we were given great feedback and possible tweaks regarding our idea which made us confident, so we began a more detailed research. AS mentioned earlier the whole concept is to expand our knowledge and improve our programming language experience so that's why we came up with this idea, framework and environments.

Our research and plan lead us to the decision of creating an android application using Android Studio (environment) using Java which will then communicate between firebase and the server, where all the face recognition is detected.

Cloud Services: The next big step of course was to decide what cloud service to use as there is so many out for example Microsoft Azure, Amazon and Google. We ended up picking Firebase which is basically owned by Google, user friendly and code friendly. We were both responsible for researching individual on what's the best cloud service, download limit, upload limit, database, data storage and then organized a library meeting discussing what we have researched and then decided the best suitable for our project.

Android Application: Android Studio is the official integrated development environment for android therefore that was what the research for android development was based on. The communication between android and server had to also be researched as it was a difficult enough task.

Server: Server is where the face recognition and motion detection runs while communicating with firebase (downloading images from data storage), authenticating users and communicating with android application.

2.2 Actual Development

2.2.1 Agile Methodology

The fact that we have learned different approaches or methods to software development we were more than happy to follow the Agile Methodology as we felt like it was most suited for us. Some of the general principles include face to face communication is the best way to transfer information, developers and businesspeople must work together to throughout entire project. The team agreed on this methodology simply because we

felt that it was a good way to approach it. This project in general was to arrange regular meetings and communication on the status or progress of each team member. Agile approaches include the division of work between team members for a short period of time called sprints between two and six weeks in length. At each beginning of the sprint a goal is set to be accomplished for the end of the sprint for example developing a feature or testing a part of the project(application). This whole approach helped us a lot with progress and showing or presenting progress to our supervisor.

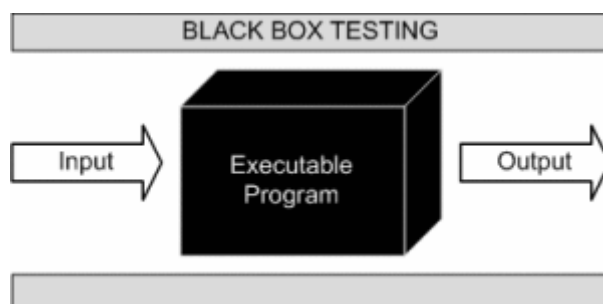
2.2.2 Development

The tasks were divided into sprints as mentioned earlier therefore now it is time to immerse ourselves into the development solution. There was a lot of issues arising along the sprints which is normal within a software project, but we faced them by communicating regularly using communication services like Facebook. We also booked a room every week to catch up on the progress or to see if there were problems to solve them as a team.

2.2.3 Testing

Through the development process there had to be a lot of testing involved if we wanted to improve our code. In our case the most common type of testing that we used was Black Box Testing **Figure 1**. Black Box Testing is also known as Behavioral Testing which is basically testing the internal structure/design/implementation o. The item being tested is not known to the user. We had to test features like moving your head left and right testing if the face recognition works accurately and if it records, frames per second etc.

Figure 1



2.2.4 Supervisor Meetings

During the first meeting with our supervisor we discussed the idea as mentioned earlier, we also discussed the frequency of each meeting, which was 1 day during the week, every week. However, if there was occasion whe we could not attend the meeting with our supervisor, we emailed him and let him know about that fact.

3. Technology Review

3.1 Introduction

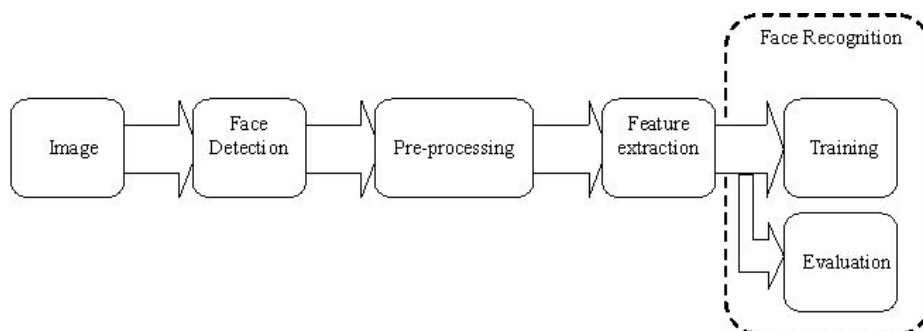
Facial recognition is an important research topic spanning numerous fields and disciplines. This is because it is used in numerous practical applications such as bankcard identification, mug shots searching, security monitoring and surveillance system therefore it is a fundamental human behavior that is essential for effective communication and interaction among people.

The great rate of development of this technology is due to a few factors: the constant or active development of algorithms, the availability of large image databases used and the performance improvement of the algorithms required [12]. In addition, a huge problem in relation to this technology involves constantly being tackled by researches from a variety of disciplines such as image processing, pattern recognition, neural networks, computer vision, computer graphics and psychology.

The basic process of face image recognition is shown in **Figure 2**. Thanks to the feature extraction after the detection of the face, and pre-processing, the key feature is further used to categorize the images, which is done with the help of a face classifier. The extracted features of the image are compared with the pictures stored in the database [13].

Even though current recognition system has reached certain levels of maturity after enormous research in the field it still has perception difficulties in relation to factors such as human poses or variation as well as facial illumination.

Figure 2



3.2 Facial Recognition system developed

The main environment used to develop the code for face recognition was an IDE called Spyder which is integrated with the python open source distribution called Anaconda. Spyder is a powerful scientific environment written in Python [14]. The IDE includes a lot of features such as an editor, I Python Console, variable explorer, profiler and a debugger; all of which are best suited for when it comes to projects that use face recognition.

3.2.1 Python: Software Programming Language for the Server



Python is an interpreted, object-oriented and high-level programming language with dynamic semantics [15]. It is commonly used in data science and in designing machine learning algorithms. This language is quick to learn and adjust to regarding the syntax, which emphasizes human readability and therefore it is easy to maintain. As a result, developers spend more time thinking about the problem they're trying to solve and less time thinking about language complexities or deciphering source code provided by others. Python has libraries that can be applied to machine learning algorithms in order to solve real problems which is another reason for choosing this language. It is both broadly supported and adopted as it is widely popular and widely used due to its high rankings in the surveys conducted by others [16].

Python was created by Guido van Rossum and first released in 1991 [17]. It supports multiple programming paradigms like procedural, object-oriented and functional programming. This language features a small library which depends on the aim that it is meant to, as described earlier on. Installations will be required if the project depends on extra libraries.

An alternative language that can be used for face recognition is Java and there are plenty of libraries out there, but there is one issue. As part of its software release (OpenCV), it offers only a few modules (with Java bindings) out of the box—and facial recognition is not one of them. Therefore, to use it, you need to manually build it [18] and because using OpenCV as a main library, for the opportunity to improve our software development

knowledge, we decided it would be a better choice overall to use Python. Otherwise, problems along the way could have arisen – errors and issues with version numbers – and by choosing OpenCV they can be avoided.

After deciding the main programming language, the next step would be picking the appropriate version of python to begin developing our project in. However, the differences between the two made it easier to pick the one which seemed most fitting.

| Python 2 | Python 3 |
|---|--|
| Many libraries are not built forward compatible | Many libraries today created are specifically created for python 3 |
| Strings are stored as ASCII | Text strings are Unicode |
| Rounds your calculation to nearest whole number | Returns exact result as expected |
| Print “Hello” | Print (“Hello”) |

3.2.2 Spyder: Integrated Development Environment (IDE)



Beyond Spyder's many built-in functions, as mentioned earlier on, its abilities can be extended even further via its plugin system and API [19]. Spyder can also be used as PyQt5 extension library, allowing you to build up its functionality and embed its components, like the interactive console for example.

Core components of the Spyder platform:

- Editor with syntax highlighting
- Support for multiple Python consoles
- Capable of exploring and editing variables from GUI
- Debugger linked to IPdb, for step by step execution
- Run-time profiler to benchmark code
- Project support, allowing multiple development efforts simultaneously
- Built in file explorer
- History Log

Microsoft Visual Studio is another platform that can be used when coding python applications. The main reason Spyder was used is because it comes integrated with Anaconda, which makes installation very easy and flexible. Anaconda is available on Windows, macOS and Linux, while Microsoft Visual Studio is only available on Windows and can be pricey. Putting that aside, the platform has its many benefits such as allowing you to store your source code, backlog and other project data via your Visual Studio Online Account, with no servers to deploy, configure or manage [20].

3.2.3 Libraries

3.2.3.1 Haar Classifier: Face Recognition Library

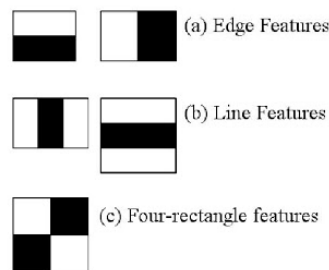
The face recognition library that was researched and put to practice is the Haar Classifier, provided by the Python library called Open Source

Computer Vision (OpenCV). The Haar Classifier is a machine learning based approach, an algorithm, created by Paul Viola and Michael Jones [21]. The algorithm needs a lot of positive images (with faces) and negative images (without faces) to train the classifier. The apparent characteristics are then extracted from it using the Haar Features as shown in **Figure 3** Error! Not a valid bookmark self-reference.Error! Not a valid bookmark self-reference..

Essentially, each characteristic represents a single value and it is calculated by subtracting the sum of pixels under the white rectangle from the sum of pixels under black rectangle. Unfortunately, as a result of the amount time it would take to process each image, integral image is introduced. This reduces the computational needs and in turn makes the detection a lot quicker. Among all the calculated features, most of them end up being irrelevant. Thus, to be able to select the best and most accurate features, Adaboost was used. It selects both the best features and trains the classifiers that use them [23].

A Cascade of Classifiers is introduced in order to focus the window on face region if it is not to discard it and not process it again, instead to focus on regions where there can be face. OpenCV comes with trainer as well as a detector therefore it comes with pre-trained classifiers for face, eyes, smiles etc. [24] .

Figure 3 [22]



3.2.3.2 OpenCV: Motion Detection Library

Background subtraction is critical in many computer vision applications. We use it to count the number of cars passing through a toll booth, the number of people walking in and out of a store and we also use it for motion detection.

There are many ways to do the motion detection using OpenCV, which can be both complicated and simple depending on the methods used. The two most common methods are forms of Gaussian Mixture Model based foreground and background segmentation. For the least CPU usage, yet reasonably effective, a way to do would be assuming that the first frame of our video stream contains the background we want model and nothing more. Therefore, background subtraction will be used to calculate the motion detection and draw a bounding box surrounding the region of the image that contains motion [25].

The disadvantages of this method are that it can easily fail due to the shadowing, reflections, lighting conditions and any other possible change in the environment. Nonetheless, for our purpose it is perfect, as we don't need to invest into utilizing fixed mounted cameras and controlled lighting conditions.

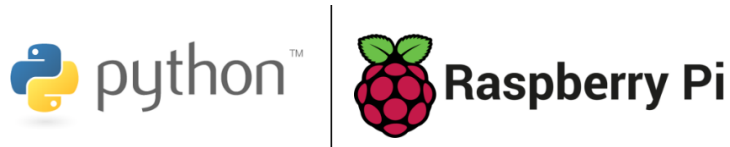
3.2.3.3 Tkinter: GUI Library

Tkinter is the library used for GUI, it has been an integral of Python for a long time. It is the standard Python interface to the Tk GUI toolkit [26]. Tkinter package is a thin object-oriented layer on top of Tcl/Tk. Tkinter is not the only GUI Programming toolkit for Python but it is the most commonly used one. You don't need to worry about installation of the Tkinter module as it comes with Python default.

3.2.4 Hardware used in Facial Recognition

In computing, hardware refers to all the physical, tangible components that take up space. Unlike software, which is related to programs necessary for the computer to run, hardware can be seen and touched. All computers, whether it's a desktop or a laptop, have some common hardware components. The only difference between the two being the size and capabilities of the individual components [27]. In this case the hardware we decided we are going to use are raspberry Pi, which acts as a surveillance system or server, and a camera to identify and detect people's faces.

3.2.4.1 Raspberry Pi



Raspberry Pi is a low cost, credit card sized computer. It plugs into a TV or computer monitor and requires a keyboard and mouse just like a normal computer. Its main purpose is to explore computing. It can be used to code in languages like Python and Scratch, and it is widely used among developing large projects like robots, cars and many others.

It is capable of performing the same tasks as a personal computer like browse the internet, play high definition videos; it works with programs like Microsoft Word, excel and so on [10]. In our case we are using it to act as a server to record the videos captured by the webcam and transfer them to the cloud service. It usually comes with a memory card with a software called NOOBS installed which basically give you an option of few operating systems available like Raspbian. This operating system is a Linux Distribution, built on the Linux Kernel [28], and it contains Java SE Platform products licensed under the Oracle Binary Code License Agreement.

After extensive research was conducted on this hardware, we decided to go ahead and use Raspbian Stretch with desktop, and the recommended software which is available on their official website [29]. This OS was built specially for the Raspberry Pi, and its latest version, Stretch, makes a considerable amount of changes to optimize performance. The reasoning behind this was due to our application being already heavy, in terms of its CPU performance. We needed to pick the best operating system to improve the overall performance.

An alternative operating system is Windows 10 IoT. Although, since we used, we used Python to design our project, Linux's friendly operating systems was an option, as Python and other common libraries are already installed for our purpose.

3.2.4.2 Camera

We used Advent AWCAMHD15 Full HD Webcam for our project due to reasons like the fact that most laptops come with integrated webcams but for our project we wanted higher accuracy when it came to the face detection therefore integrated laptop webcam wasn't a guarantee that they will match the quality that we needed. Advent webcam also came with 1920 x 1080 resolution, 30 frames per second and very good lens therefore overall was a great choice in our project and improved the accuracy [30]. Installation was very simple with USB 2.0 port , just connect and driver installation was automatic.

3.3 Android application development

More than half of the world's population owns a cell phone, more specifically 66.72% [31] of the world. 79% of people between the ages of 18-44 have their smartphones with them at least 22 hours a day [32], which accounts for the vast majority, while 63% keep them near for 21 hours a day [32].

This is proof that most of the population cannot go anywhere without their phones for long periods of time. From this research we have concluded that using an application on the user's device to allow people to keep an eye on the house is the best approach.

When people are away from home there is nobody there to make sure no burglary will take place. This is where this application comes in. With the help of facial recognition program developed, the user can receive notifications about who was detected in the camera. If they wish, they can choose to download the videos of that person with timestamps on them, to make sure no crime was committed.

3.3.1 Java: Software Programming Language for the Client



Java was originally called Oak [33] – which was then changed to Green and finally became Java – was developed by James Gosling, Mike Sheridan and Patrick Naughton and was initially designed for interactive television. This object-oriented programming language was eventually released in 1995 as a “core component of Sun Microsystems’ Java Platform” [34].

Most languages are using the compile-link-execute format. The compiler converts the source code into low-level language, other languages use interpreters to take the high-level statements, figure out the low-level version and then runs the output. Java combines of those methods as supposed to doing them separately [35],

Java programming language is “accepted as one of the best programming languages” [36] in relation to mobile application development because it is easy to learn and human readable, it has open source libraries and it contains a vast amount of API’s [37].

There are other languages that can be used for android applications, but Java has the most support from programming communities and it is one of the most popular programming languages used. It is the 2nd most popular language in the world according to the PPYT Index [38].

3.3.2 Android Studio: Integrated Development Environment (IDE)



An integrated development environment is a software application used for software development. Most IDE’s consist of a source code editor,

build automation tools and a debugger [39]. The one we have decided to use for the mobile application is Android Studio because of the numerous benefits it provides [40]:

- Faster deployment of fresh builds – the instant run can test the code on an android emulator provided by the platform.
- Accurate and programming – thanks to the IntelliJ IDEA interface with which the Code Editor is equipped, code analysis is currently more accurate and a lot easier to implement.
- Fast programming and testing – the new emulator is three times faster than its previous one and it's also quicker than a real android device.
- Better app indexing – Android Studio provides an App Indexing feature which is used to create indexable URL links to the mobile application.

However, Android Studio isn't the only framework that can be used to develop android applications and making a choice was indeed, a challenge. Besides choosing it as part of our learning curve, there are other reasons that made the decision final.

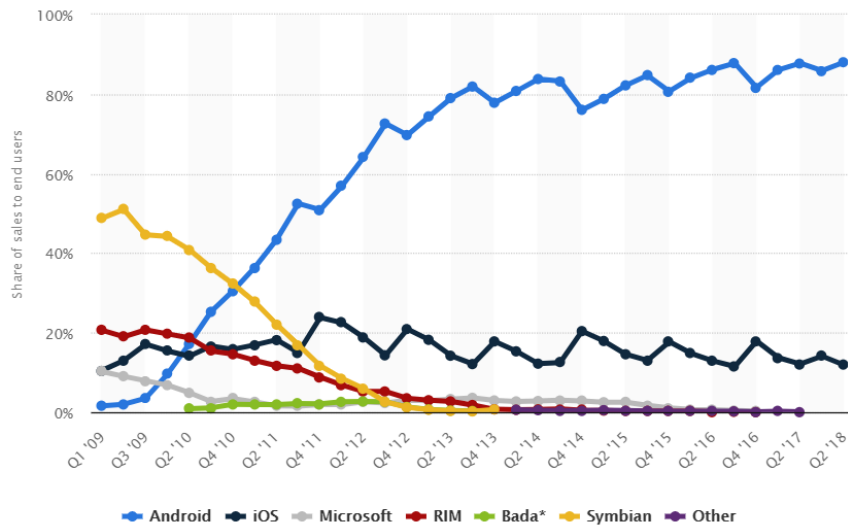
Ionic is an open source SDK, created by Ben Sperry and Adam Bradley in 2013 [41] and it is used to design and develop hybrid mobile applications. Its focus is mainly on the user experience which is why it can also be described as a UI toolkit. It uses a vast majority of technologies including HTML, CSS, JavaScript and Angular JS

| Ionic Framework [42] | | Android Studio | |
|--|---|---|---|
| Advantages | Disadvantages | Advantages | Disadvantages |
| It is open source | Compared to native applications, Ionic lacks performance | An open source platform | The performance exceeds that of the Ionic Framework |
| Apps developed with this framework work on multiple platforms including Android, | Incapability to provide security is a huge drawback as it is crucial in preventing people from accessing personal | Security features such as encrypted file system to protect user data when the smart phone is either | Unlike Ionic, this is not a cross-platform framework and it only runs on android applications |

| | | | |
|--------------------------------------|---|--|---|
| iOS and Windows (mobile and desktop) | information | stolen or lost are build-in [43] | |
| Availability of readily made plugins | From my experience with Ionic, size is an issue; especially when using a computer without enough space to support the application development | There are many plugins already provided by the framework | Compared to Ionic, when this framework does not occupy as much space while a program is being developed |

Android studio was developed on JetBrains IntelliJ IDEA for the Google Android operating system its first release being version 0.8 in June 2014, while the stable 1.0 version being in the last month of the same year [44]. From there, this platform started becoming very popular starting from Google phones to Samsung, LG and HTC as they could not build one of their own due to lack of resources, time and community support [45]. Although it is true that Android Studio does not support multiple platforms, Android is still is the most popular operating system used in smartphones as shown in **Figure 4** . That means that building an application for Android is not a huge disadvantage to the users.

Figure 4 [46]



3.4 Firebase: Cloud Service Provider



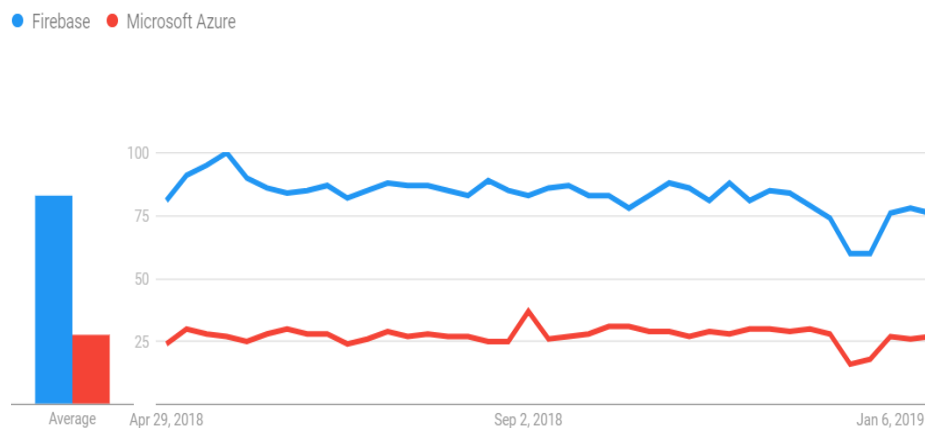
Cloud computing “was used to represent networks of computing equipment” in between 1977 and 1981. The term “cloud” was originally used as a representation of the platforms for distributed computing [47]. A communication service provider (CSP) is a third-party company that offers cloud computing-based services to businesses or individuals [48]. These services are split into three main categories [49]:

1. **Infrastructure as a service (IaaS):** such as virtual servers, virtual storage and virtual computers.
2. **Software as a service (SaaS):** it delivers both simple and complex software via the internet.
3. **Platform as a service (PaaS):** a service created as a result of combining both IaaS and SaaS.

Firestore was originally called Envelope and it was established by James Tamplin and Andrew Lee back in 2011. It is a mobile and web application development platform developed by Firebase which was formally launched in 2012 [50]. This was one of our main cloud service choices from a variety of options as the platform provides numerous benefits such as real-time database, storage, authentication, cloud messaging and notifications, free usage and so on [51].

Another very good cloud computing platform we considered using is Microsoft Azure previously known as Windows Azure. This is an ever-expanding set of cloud services, released by Microsoft and in 2010 with the purpose of “building, testing, deploying and managing applications and services” [52]. Using Azure with Android applications is a lot less common as shown in **Figure 5**, thus there isn’t enough support on online communities to help with issues that could arise.

Figure 5 [53]



3.4.1 Firebase Cloud Service Uses

3.4.1.1 Image and Video storage

The android app is first and foremost used to upload images of the customer’s acquaintances to the cloud storage and database and stored in a Gallery in the user’s device from the database.

The Raspberry Pi surveillance system downloads those from Firebase storage, unlike android which requires the creation of a database to be able

to download any files. That way, the system has access to the data required to train itself, thus allowing it to differentiate between a known face and an unknown one.

Once motion detection identifies people present in front of the camera, it records a 10 second video that is then uploaded to the cloud storage as well as the database. As the mobile application from the customer's device is linked to the same cloud, there is an ease in accessing those videos by downloading them directly from the application the same way we store the images in a Gallery, via the database. However, in this case the videos are downloaded to the device's internal memory to prevent loss of evidence in case the device is stolen or lost.

The program allows the user to download videos related to a stranger seen on camera as well as someone they know. In relation to their acquaintance, the full name needs to be entered in the pop-up window to let the cloud service know whose videos they wish to see.

3.4.1.2 Authentication

As most apps require some sort of user identity to enhance security in relation to personal information, Firebase provides "backend services, easy-to-use SDKs, and ready-made UI libraries Authentication" to. It can be used with passwords, phone numbers or providers such as Google or Facebook.

This is crucial in preventing people to access the images and videos captured by the surveillance camera. If someone outside the household location were to access the user's storage, the program would have been a major cause of privacy violation.

3.4.1.3 Pusher Beams: Sending Notifications from the Server to the User

Pusher Beams is a library product which allows real-time notification deliverability across different technologies and locations. Push notifications are crucial to keep the customers up to date on deliveries, or in our case, home security.

Unlike most mobile tools, Beams does not send automatic notifications. Instead, it allows developers to program a trigger push notification based on the user's in-app activity [54]. It is free and unlimited, unless the devel-

opers wish to support users above the given quota of 1,999 active devices. This library not only is easy to use and readily available, but it was engineered to be considerate of user's privacy, who can opt-out at any time.

This library contains a database of tokens provided by APNs on iOS and FCM on Android which update automatically. This benefits the developers as they do not need to consistently update them manually.

Within the Beams API, there is a that allows the developers to filter the devices that receive notifications. This feature is called Interests and the way it does it is by splitting them into different categories. This API is so fast, that it can deliver up to 1 million notifications every 17 seconds in Android, which is about 58,823 notifications a second.

The analytical side of the API gives way to an acknowledgement event to be reported every time the mobile device receives a notification and every time the user opens that notification; it reports back an open event. This means that developers can see watch users open the notifications in real-time via the Beams Debug Console. [55].

3.5 Frameworks used

3.5.1 Windows 10

The first operating system ever created by Microsoft dates to 1985, when Windows 1.0 was created. After that, almost every year a new version was release [56]. Currently, the oldest version of Windows that is still widely used is XP as seen in despite being nearly 18 years old. This shows how vastly popular this operating system is, also shown in **Figure 6** .

Figure 6 [57]



However, since 2015, Windows 10 became the newest of the Microsoft Windows operating system line. Some newly introduced updates include a better taskbar notification center, virtual desktops support and Microsoft's virtual assistant, Cortana [58].

There are two Windows 10 Editions available to the consumers; Windows 10 Pro and Windows 10 Home. There have been several additions of Windows 10 editions such as Windows 10 Mobile, Windows 10 Enterprise, Windows 10 Education, but unfortunately, these are not available directly to the users. The ones we have used to build the camera security system are the Windows 10 Pro and Education.

3.5.2 Raspbian

Raspbian is a Debian-based computer OS originally created in 2012 by Mike Thompson and Peter Green as an independent project with the Linux Kernel. In 2015, this software has received an official approval from the Raspberry Pi Foundation, to use it as their core OS. It has various versions, including Raspbian Stretch – which is the one we are using for our Raspberry Pi – as well as Raspbian Jessie.

Starting the latest update, it uses Pi Improved X-Window Environment, Lightweight as its desktop environment which is composed of an LXDE desktop environment as well as the Openbox stacking window manager [59].

3.5.3 Android

Android is a mobile operating system developed by Google and it started off from a modified version of the Linux Kernel. The software comprises of middleware and key applications on top of the operating system. It is a powerful OS which supports applications such as Smart Phones and Tablets [60]. Most of the android hardware are based on the ARM architecture platform [61]

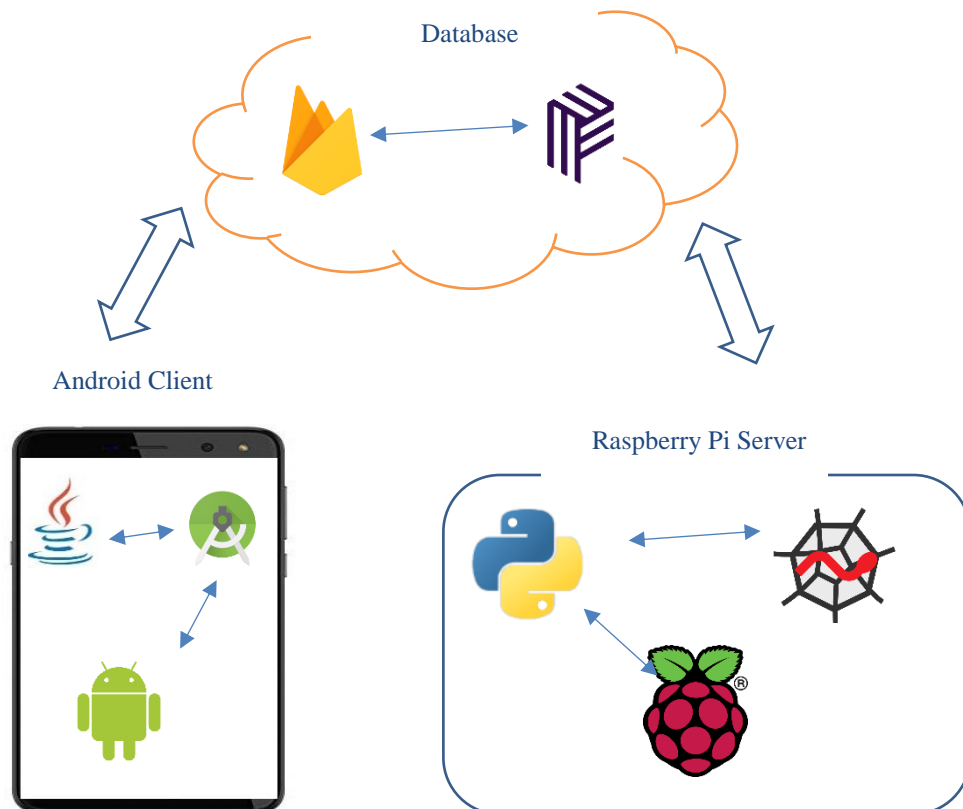
Some of its features and specifications include mobile devices optimization, an open source Web engine and the enabling of reuse and replacement of components. We all know that android is the most used operating system in the world, but not many people know how many companies support Android in their hardware. There are 22 such brands including: Alcatel, Acer, HTC, Huawei, Lenovo, LG, Motorola, Samsung, Sony Ericsson and OnePlus.

4. System Design

This application consists of two parts, (1) the client and (2) the server. The client is the android application that lets the user know if someone came to visit, and whether they know that person or not, while the server is the surveillance camera system linked to the Raspberry Pi that's dealing with the image recognition and motion detection. In this section we will lay out the Architecture of the program as well as a detailed explanation of its core functionalities.

4.1 Architecture

This section will cover the project's overall design and architecture. The core brain of this project is the database, and it allows for client-server communication involving notifications, videos, images, user information and so on. To thoroughly explain each part in this project we will be revealing screenshots and code snippets of both the client and the server.



4.2 Database Design

The design of the database can be seen in the above architecture. The main development features Firebase provides that we have been using are the Authentication, the Database and the Storage.

The user first must create an account via the mobile application which is then stored in the Firebase Authentication as shown in **Figure 7** and its Realtime Database **Figure 8** – which is not necessarily required to be able to log into both programs. However, creating a Realtime database for the user's data creates a Unique Identifier (UID) allows privacy protection and the storage of the user's name. In other words, no other user can access the information unless their UID matches the one from the database.

Figure 7


| Identifier | Providers | Created | Signed In | User UID ↑ |
|------------------------|---|--------------|--------------|------------------------------|
| stoyanrizov6@gmail.com |  | Feb 15, 2019 | Apr 25, 2019 | aFe4RebZjtcvvaboqqqbZNDyAQi2 |

Figure 8

```
... users
...
... aFe4RebZjtcvvaboqqqbZNDyAQi2
...
... email: "stoyanrizov6@gmail.co
... name: "Stoyan Rizov
```

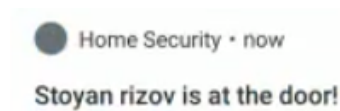
The design of the database can be seen in the above architecture. The main development features Firebase provides that we have been using are the Authentication, the Database and the Storage.

Firebase is also the bridge that allows the access of images and videos. The user uses the mobile application to upload pictures of family members and friends to the cloud. The server then takes those images, download them and uses them to train the data. This in turn allows the server to rec-

ognize the people seen on the camera by looking up the trained data and comparing it to the face of that person. If images of that person have been uploaded to the cloud, that person's name will be recognized, otherwise it will be unknown.

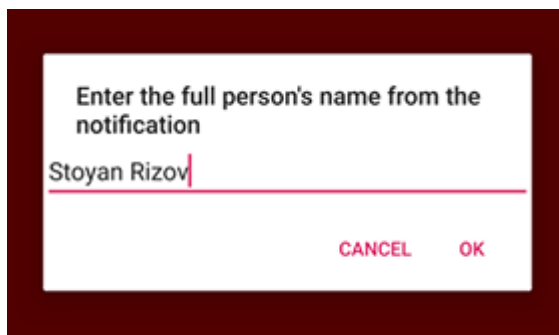
A notification will then be sent to the user, via Pusher Beams (as shown in the database section of the architecture). If the visitor is a known family member, or friend, their name will appear in the notification as "<person name> is at the door" as seen in **Figure 9** . In case of an intruder, "unknown" will appear instead of the person's name.

Figure 9



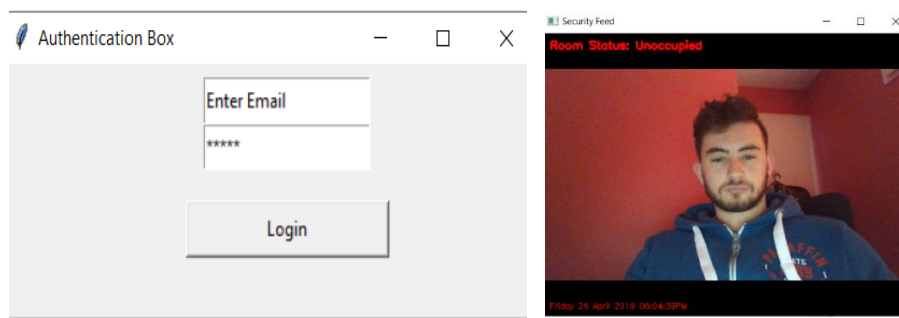
Once the motion has been detected on the camera, the recording starts. A 10 second video is captured and uploaded to the cloud. Once that was done, the customer can choose whether they wish to access the videos. If the notification shows a family member has visited, the customer can enter the name in a pop-up window – **Figure 10** – and the video to download all the videos recorded of that person. Otherwise, need only press a button specially designed to download all the unknown visitors recorded.

Figure 10



4.3 Server Design and Functionality

The design of the server is simply configured with user authentication provided by the library tkinter. The user can enter their email and password which are stored in the database in the cloud service (firebase). If the user does not enter correct information, they won't be capable of using the service. If the user is authenticated then their information is downloaded like images that are stored and used as face recognition.



Motion detection is another feature into the server-side application, as shown above a window is displayed until there is movement, once there is movement window is closed and face recognition opens in the background and starts recording for a set time (10 seconds) as outlined in other chapters. Motion detection window displays current date and time and room status whether there is current movement or not. Video after being recorded is sent to the firebase, where it is stored in the data storage, can be downloaded by the android application and accessed when authenticated on the mobile application.

4.4 Client Design and Functionality

The client is the android application with which the user interacts. The user must first register to be able to use the surveillance system. In our case, the users can only access the images uploaded from their mobile phones and therefore avoiding any privacy related issues.

3.5.4 Registration

The login page is the first page that runs when the application is opened. It provides the user with the choice of logging in or registering as shown in **Figure 12**. If the customer does not own an account, they will require one

to be able to use the Raspberry Pi system, which is where the Registration page come in. Once the user presses the “Register” button, they are brought to the page shown in **Figure 11**. Once registered, they can proceed to log into both, the mobile application and the server application.

Figure 12

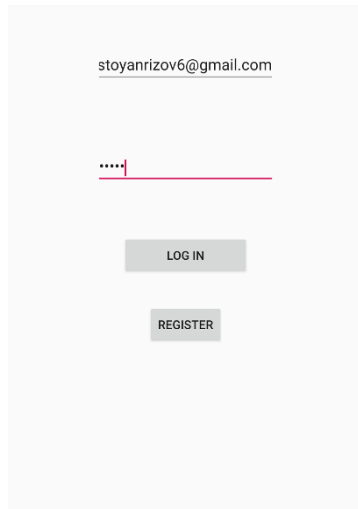
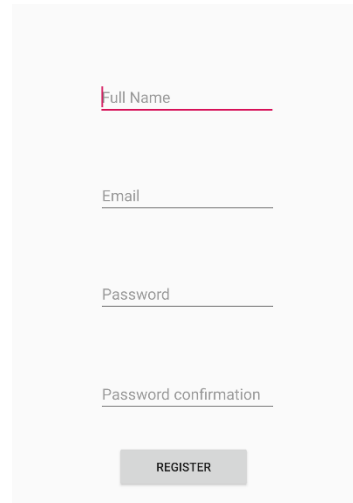
A login form interface with a light gray background. It features two input fields: the first contains the email address 'stoyanrizov6@gmail.com' and the second contains a masked password '.....'. Below these fields are two buttons: 'LOG IN' and 'REGISTER', both with a light gray background and black text.

Figure 11

A registration form interface with a light gray background. It features four input fields: 'Full Name', 'Email', 'Password', and 'Password confirmation'. Below these fields is a single 'REGISTER' button with a light gray background and black text.

The registration must be complete. In other words, all fields must be filled in, otherwise the registration will not be successful. Apart from that, the password and the confirmation password fields must match as well. Once these rules are followed, the registration will be successful.

3.5.5 Login

The application provides a log in functionality – – so that the customer can enjoy all the features our application has to provide. The user must fill in all the fields provided, the same way the registration is done. The email a password must match the ones from the database, or the action will be unsuccessful.

3.5.6 Home

This is the page with all core functionality of the application. Once the user is logged in, he or she is brought to the home page, Error! Reference source not found.

To upload an image to the cloud, the user must enter the name of the person whose image they wish to upload and choose out of the two options provided in Error! Reference source not found. taking a photograph or choosing one from the gallery. Whenever the choice is made, the image will be shown in the frame shown in Error! Reference source not found.. If the customer is happy with the image and they wish to upload it, they can press on the upload button – Error! Reference source not found..

Figure 15



Figure 14



Figure 13



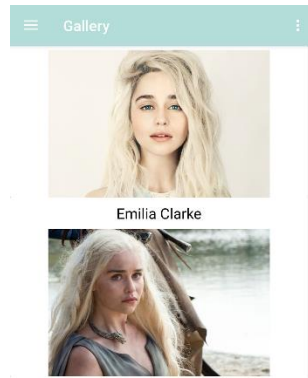
Figure 16



Once
the images
are

uploaded, they can be viewed in the application Gallery with the name of the person provided above each image – **Figure 17**. Every user has a separate gallery according to the images they have uploaded from the application.

Figure 17



The user can also download videos as mentioned earlier by choosing one of the two buttons shown in **Figure 19**, from the dropdown menu and call 911 with the button in **Figure 18** if the videos are showing criminal activity as well as log out – **Figure 20**.

Figure 19

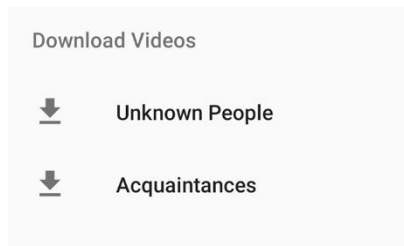


Figure 18

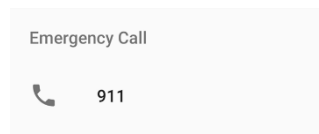


Figure 20



5. System Evaluation

In this chapter we will discuss both the Android application and the server application based on different attributes. Both applications will be

evaluated individually and as well as interconnected applications. The evaluation will be based on these attributes:

- Robustness and Efficiency
- Scalability
- Security and Validation

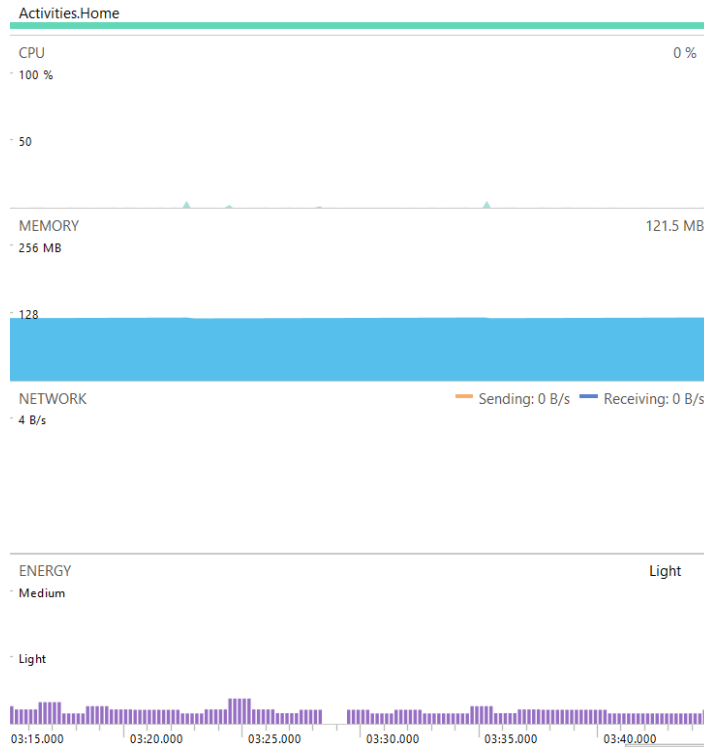
5.1 Robustness and Efficiency

Server Side: When dealing with python and face recognition it is quite difficult to minimize CPU or GPU usage as it is in general a very demanding process. However, there are few tweaks and benefits that you could implement to run the application better. The obvious one is having a good GPU and CPU power. Another one is making sure when running the application, if you are using Python 3.4 or newer, pass in a ‘--cpus parameter’ [62]. Anyways, we did go ahead with OpenCV for the simple reason that it was the least performance demanding and adaptable for Raspberry Pi.

Android application: Android has an integrated tool called Profiler [63] that provides real-time data to help figure out the amount of CPU, memory, network and battery usage the application has.

As we can see in **Figure 21**, the application does not exceed 150MB in terms of memory, and the CPU, Battery and Network usage are from no usage to low. This proves that in terms of robustness and efficiency, this application’s performance is extremely high. The reason for that is the fact that most of its features do not require extreme CPU usage like a game would for example. Also since it does not run constantly, the battery usage tends to barely exist.

Figure 21



5.2 Scalability

Server Side: When it comes to scalability on the server-side, it is one per house therefore one user is allowed per house for the reason being of our general design architecture. There is one user, like an admin in one house and then other users in the house are counted as sub-users therefore there is no need for multiple logins at a time, improving the performance and not slowing it down nor slowing down the accuracy throughout the day.

Android application: In terms of storing images and videos, Firebase does have its limitations when creating a free account. As this project is based on the Firebase storage as well as the real time database storage for a considerable amount of data, the limitations it provides are a problem.

For real time storage, in relation to syncing the images and videos between the client and the server, it only allows a storage of 1 GB and down-

loading only 10 GB worth of files per month. The original storage on the other hand allows 5 GB of storage while permitting a daily download of 1 GB only.

If this project is to be further developed on a more advanced level, it would certainly require a paid Firebase account to rid of those limitations

5.3 Security and Validation

Server Side: Security on the server side is very simple as we mentioned before, in the previous section, there is one admin per household. Therefore, the admin of the house downloads the folder related to the house. However, the main user has no access to other users in the house so there is privacy among other customers. Below is a console output of what happens when the customer enters incorrect information into the authentication box.

```
HTTPError: [Errno 400 Client Error: Bad Request for url: https://www.googleapis.com/identitytoolkit/v3/relyingparty/verifyPassword?key=AIzaSyBpRsAequKTjY25_ew-RutT31eE4COHU9E] {
  "error": {
    "code": 400,
    "message": "INVALID_EMAIL",
    "errors": [
      {
        "message": "INVALID_EMAIL",
        "domain": "global",
        "reason": "invalid"
      }
    ]
  }
}
```

Android Application: In terms of security, as I mentioned earlier, for the user to access the images uploaded to the cloud as well as all the videos captured by the camera security system, they are required to be registered and logged in. Android also provides an encrypted file system that protects user data when the mobile is stolen or lost [43]. This gets rid of any worry that someone who stole the phone may have a hold of the user data within.

6. Conclusion

To conclude this project, our team has developed a fully working Webcam Security System that can record a video when motion is detected and faces that are in the video are recognized, using OpenCV. The recognized faces are then sent to the cloud service system we used, called Firebase, including the video where it is then accessed using an android application.

Our focus and motivation were good as we were able to develop the final product as planned. However, there we did run into a few difficult problems along the way and improvements that could have been made.

The project led us to accomplish a lot including improving programming languages, our experience in working in a different environment, planning, working as part of a team and meeting a set deadline given in time.

One improvement that could be made for the server side is figuring out a way to run the motion detection simultaneously with the face recognition. Nonetheless, when I tried this, using OpenCV. We found out that including multi-threading was not possible as CPU usage would jump to 100% and slow down the application completely.

If we were to develop the project again, we would have started directly from Raspberry Pi as we thought transferring from windows to Raspbian would not take as long as it did. A lot of libraries had to be manually installed where with windows it was a lot easier. Therefore, it would have been a lot more convenient to use Raspberry Pi to start off.

Another change or improvement that we would make is, instead of the user having to type up the name of the visitors to download the videos from the firebase, the user would have a drop down list with a selectable choice. Nonetheless, this can be a part of future improvements.

7. Appendices

7.1 Github Link

This is the link to our GitHub repository where the source code, screen-cast, and the README can be found:

<https://github.com/cristina1997/Main-Project-App>

8. References

- [Digital Desk staff, "Almost half of all reported burglaries in 2018 occurred in Dublin," 27 March 2019. [Online]. Available: <https://www.irishexaminer.com/breakingnews/ireland/almost-half-of-all-reported-burglaries-in-2018-occurred-in-dublin-913656.html>.
- [Irish Examiner, "Crime in Ireland: A snapshot of crime in every county," n.d.. [Online]. Available: <https://www.irishexaminer.com/crime/crimebycounty/>.
- [Central Statistics Office, "Crime - Recorded Crime Offences; Recorded Crime Offences (Number) by Type of Offence and Year," n.d.. [Online]. Available: <https://www.cso.ie/multiquicktables/quickTables.aspx?id=cja01>.
- [DB-Engines, "DB-Engines Ranking," April 2019. [Online]. Available: <https://db-engines.com/en/ranking>.
- [Technopedia, "Image Recognition," n.d.. [Online]. Available: <https://www.techopedia.com/definition/33499/image-recognition>.
- [P. Pandey, "Datacamp," Datacamp, 20 December 2018. [Online]. Available: <https://www.datacamp.com/community/tutorials/face-detection-python-opencv>.
- [Pusher, "We love Developers," Pusher, n.d.. [Online]. Available: <https://pusher.com>.
- [Pusher, "Notifications that Deliver," <https://pusher.com/beams>, n.d.. [Online]. Available: <https://pusher.com/beams>.
- [Pusher Docs, "Channels API Libraries," n.d.. [Online]. Available: https://pusher.com/docs/channels/channels_libraries/libraries.
- [Raspberry Pi, "What is a Raspberry Pi?," n.d.. [Online]. Available: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>.
- [Wikipedia contributors, "List of Linux adopters," 31 March 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=List_of_Linux_adopters&oldid=890370673.
- [A. & E.-B. A. & E.-H. A. Tolba, "Face Recognition," Ahmed A El-Harby, Mansoura University, 2005.
- [A. Rosebrock, "OpenCV Face Recognition," pyimagesearch, 24 September 2018. [Online]. Available: <https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>.
- [Spyder, "Overview," Spyder, n.d.. [Online]. Available: <https://www.spyder-ide.org/>.
- [Python, "What is Python? Executive Summary," Python, n.d.. [Online]. Available: <https://www.python.org/doc/essays/blurb/>.

- [S. Yegulalp, "What is Python? Everything you need to know," InfoWorld, 16] June 2018. [Online]. Available: <https://www.infoworld.com/article/3204016/what-is-python.html>.
- [Wikipedia contributors, "Python (programming language)," Wikipedia, The 17] Free Encyclopedia., [Online]. Available: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)). [Accessed 4 April 2019].
- [M. Bansal, "Facial recognition using OpenCV in Java," freeCodeCamp, 25 18] June 2018. [Online]. Available: <https://medium.freecodecamp.org/facial-recognition-using-opencv-in-java-92fa40c22f62>.
- [Spyder, "PyPi," Spyder, n.d.. [Online]. Available: 19] <https://pypi.org/project/spyder/?fbclid=IwAR2Plrk1QYk3L4LqsPkV-PhJxnGaC60rxuWySewYvqlcy4mUWwNSSmEvFLo>.
- [Slant, "Visual Studio with the Python Tools extension vs Spyder," Slant, n.d.. 20] [Online]. Available: https://www.slant.co/versus/1242/1246/~visual-studio-with-the-python-tools-extension_vs_spyder.
- [W. Berger, "DEEP LEARNING HAAR CASCADE EXPLAINED," [Online]. 21] Available: <http://www.willberger.org/cascade-haar-explained/>.
- [doxygen, "Face Detection using Haar Cascades," 4 July 2018 . [Online]. 22] Available: https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html.
- [OpenCV, "Face Detection using Haar Cascades," [Online]. Available: 23] https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html.
- [Wikipedia, "Haar-like feature," [Online]. Available: 24] https://en.wikipedia.org/wiki/Haar-like_feature.
- [A. Rosebrock, "Basic motion detection and tracking with Python and 25] OpenCV," pyimagesearch, 25 May 2015. [Online]. Available: <https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>.
- [Wikipedia contributors, "Tkinter," Wikipedia, The Free Encyclopedia., 25 26] February 2019. [Online]. Available: <https://en.wikipedia.org/wiki/Tkinter>.
- [Wikipedia contributors, "Computer Hardware," Wikipedia, The Free 27] Encyclopedia., 26 April 2019. [Online]. Available: https://en.wikipedia.org/wiki/Computer_hardware.
- [mpthompson, "HISTORY: The start of raspbian (Debian Hard Float (armhf) 28] for RPi)," Raspberry Pi, 20 March 2012. [Online]. Available: <https://www.raspberrypi.org/forums/viewtopic.php?f=66&t=4256>.
- [Raspberry Pi Starter Kits, "Best Operating Systems for Raspberry Pi 2 & 3," 29] Raspberri Pi Starter Kits, [Online]. Available: <https://www.raspberrypistarterkits.com/products/operating-systems-raspberry-pi/>.
- [PCWORLD, "ADVENT AWCAMHD15 Full HD Webcam," [Online]. 30] Available: <https://www.pcworld.co.uk/gbuk/computing-accessories/computer->

accessories/webcams/advent-awcamhd15-full-hd-webcam-10119647-pdt.html.

[Bankmycell, "How many people have phones in the world?," Bankmycell,
31] n.d.. [Online]. Available: <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>.

[A. Stadd, "79% Of People 18-44 Have Their Smartphones With Them 22
32] Hours A Day [STUDY]," adweek, 2 Apr; 2013. [Online]. Available:
<https://www.adweek.com/digital/smartphones/>.

[Wikipedia contributors, "Oak (programming language)," Wikipedia, The Free
33] Encyclopedia., 5 March 2019. [Online]. Available:
[https://en.wikipedia.org/w/index.php?title=Oak_\(programming_language\)&oldid=886257341](https://en.wikipedia.org/w/index.php?title=Oak_(programming_language)&oldid=886257341).

[Wikipedia contributors, "Java (programming language)," Wikipedia, The Free
34] Encyclopedia., 23 April 2019. [Online]. Available:
[https://en.wikipedia.org/w/index.php?title=Java_\(programming_language\)&oldid=893797610](https://en.wikipedia.org/w/index.php?title=Java_(programming_language)&oldid=893797610).

[Jcarroll, "How java works," Carnegie Mellon University, School of Computer
35] Science, n.d.. [Online]. Available: <http://www.cs.cmu.edu/~jcarroll/15-100-s05/supps/basics/history.html>.

[A. Nordeen, "Why Java is the best programming language for Mobile
36] Applications," Irish Tech News, 19 November 2018. [Online]. Available:
<https://irishtechnews.ie/why-java-is-the-best-programming-language-for-mobile-applications/>.

[R. Limbasiya, "Which is the best language for mobile app development?,"
37] Quora, 17 June 2016. [Online]. Available: <https://www.quora.com/Which-is-the-best-language-for-mobile-app-development>.

[S. Times, "Top Computer Languages," Statistic Times, 16 February 2019.
38] [Online]. Available: <http://statisticstimes.com/tech/top-computer-languages.php>.

[Wikipedia contributors, "Integrated development environment," Wikipedia,
39] The Free Encyclopedia., 23 April 2019. [Online]. Available:
https://en.wikipedia.org/wiki/Integrated_development_environment.

[Root Info Solutions, "Top 5 Benefits from Android Studio 2.0 for an Android
40] App Development Company," Root, 19 May 2016. [Online]. Available:
<https://rootinfosol.com/top-5-benefits-from-android-studio-2-0-for-an-android-app-development-company>.

[Wikipedia contributors, "Ionic (mobile app framework)," Wikipedia, The Free
41] Encyclopedia., 17 April 2019. [Online]. Available:
[https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)).

[M. Panchal, "Ionic vs Native Apps : Revealing an Outstanding Platform for
42] your Mobile App," Excellent Webworld, n.d.. [Online]. Available:
<https://www.excellentwebworld.com/ionic-vs-native-app-development/>.

[Android Developers, "Security tips," Android Documentation, n.d.. [Online].
43] Available: <https://developer.android.com/training/articles/security-tips>.

- [Wikipedia contributors, "Android Studio," Wikipedia, The Free Encyclopedia.,
44] 19 April 2019. [Online]. Available:
https://en.wikipedia.org/w/index.php?title=Android_Studio&oldid=893154780.
- [A. Agarwal, "How did Android get so popular?," Quora, 19 June 2015.
45] [Online]. Available: <https://www.quora.com/How-did-Android-get-so-popular>.
- [Statista, "Global market share held by the leading smartphone operating
46] systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018," Statista,
2019. [Online]. Available: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.
- [Wikipedia contributors, "Cloud computing," Wikipedia, The Free
47] Encyclopedia., 25 April 2019. [Online]. Available:
https://en.wikipedia.org/wiki/Cloud_computing.
- [M. Rouse and S. Smith, "Cloud service provider (cloud provider)," Tech
48] Target, [Online]. Available:
<https://searchitchannel.techtarget.com/definition/cloud-service-provider-cloud-provider>.
- [Technopedia, "Cloud Provider," Technopedia, n.d.. [Online]. Available:
49] <https://www.techopedia.com/definition/133/cloud-provider>.
- [Wikipedia contributors, "Firebase," Wikipedia, The Free Encyclopedia, 19
50] April 2019. [Online]. Available: <https://en.wikipedia.org/wiki/Firebase>.
- [cmarix, "14 Benefits of using Firebase," cmarix, 29 August 2016. [Online].
51] Available: <https://www.cmarix.com/14-benefits-of-using-firebase/>.
- [Wikipedia contributors, "Microsoft Azure," Wikipedia, The Free
52] Encyclopedia., 22 April 2019. [Online]. Available:
https://en.wikipedia.org/wiki/Microsoft_Azure.
- [Stackshare, "Firebase vs. Microsoft Azure," Stackshare, n.d.. [Online].
53] Available: <https://stackshare.io/stackups/firebase-vs-microsoft-azure>.
- [Pusher, "Pusher Docs Beams," Pusher, n.d.. [Online]. Available:
54] <https://pusher.com/docs/beams>.
- [M. Hughes, "Pusher's new Beams API wants to make push notifications suck
55] less," The Next Web, September 2018. [Online]. Available:
<https://thenextweb.com/dd/2018/09/06/pushers-new-beams-api-wants-to-make-push-notifications-suck-less/>.
- [Wikipedia contributors, "List of Microsoft Windows versions," Wikipedia,
56] The Free Encyclopedia., 25 April 2019. [Online]. Available:
https://en.wikipedia.org/wiki/List_of_Microsoft_Windows_versions.
- [statcounter, "Desktop & Mobile Operating System Market Share Worldwide,"
57] statcounter, n.d.. [Online]. Available: <http://gs.statcounter.com/os-market-share/desktop-mobile/worldwide/#monthly-201901-201903-map>.
- [T. Fisher, "Microsoft Windows 10," Lifewire, 29 February 2019. [Online].
58] Available: <https://www.lifewire.com/windows-10-2626217>.

- [Wikipedia contributors, "Raspbian," Wikipedia, The Free Encyclopedia., 26
59] April 2019. [Online]. Available: <https://en.wikipedia.org/wiki/Raspbian>.
- [E. Garage, "Introduction To Android," Engineers Garage, n.d.. [Online].
60] Available: <https://www.engineersgarage.com/articles/what-is-android-introduction>.
- [A. Thakur, "ARM (Advanced RISC Machines) Processors," Engineers Garage,
61] n.d.. [Online]. Available: <https://www.engineersgarage.com/articles/arm-advanced-risc-machines-processors>.
- [StackOverflow, 14 August 2018. [Online]. Available:
62] <https://stackoverflow.com/questions/51842495/python-face-recognition-slow>.
- [A. Studio, "Measure app performance with Android Profiler," Android Studio,
63] n.d.. [Online]. Available: <https://developer.android.com/studio/profile/android-profiler?hl=en>.