TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Major Project Report

On

IoT Based Smart Mirror with Face Recognition

Submitted By:

Dipesh Lamichhane (072/BEX/318)

Prabin K. Shrestha (072/BEX/326)

Shishir Pantha (072/BEX/339)

Shreekar Tiwari (072/BEX/340)

Submitted To:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

THAPATHALI CAMPUS

KATHMANDU, NEPAL

Under the Supervision of

Binod Sapkota

July, 2019

TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Major Project Report

On

IoT Based Smart Mirror with Face Recognition

Submitted By:

Dipesh Lamichhane (072/BEX/318)

Prabin K. Shrestha (072/BEX/326)

Shishir Pantha (072/BEX/339)

Shreekar Tiwari (072/BEX/340)

Submitted To:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
THAPATHALI CAMPUS
KATHMANDU, NEPAL

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE BACHELOR'S DEGREE

IN ELECTRONICS AND

COMMUNICATION ENGINEERING.

Under the Supervision of

Binod Sapkota

July, 2019

DECLARATION

We hereby declare that the report of the project entitled **"IoT Based Smart Mirror with Face Recognition"** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Dipesh Lamichhane (072/BEX/318) _____

Prabin K. Shrestha (072/BEX/326) _____

Shishir Pantha (072/BEX/339) _____

Shreekar Tiwari (072/BEX/340) _____

Date: July, 2019

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled "**IoT Based Smart Mirror with Face Recognition**" submitted by **Dipesh Lamichhane, Prabin K. Shrestha, Shishir Pantha** and **Shreekar Tiwari** in partial fulfillment for the award of Bachelor's Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor's degree of Electronics and Communication Engineering.

_____

Project Supervisor

Binod Sapkota

Department of Electronics and Computer Engineering, Thapathali Campus

_____

External Examiner

_____

Project Coordinator

Dinesh Baniya Kshetri

Department of Electronics and Computer Engineering, Thapathali Campus

_____

Er. Kiran Chandra Dahal

Head of the Department,

Department of Electronics and Computer Engineering, Thapathali Campus

July, 2019

COPYRIGHT

ACKNOWLEDGEMENT

It is a pleasure to remind the fine people in the Department of Electronics and Computer Engineering of Thapathali Campus, IOE for their sincere guidance that we students received to perform our projects successfully. First, we would like to thank the Head of our Department for providing this platform to explore our knowledge and skills practically and providing us with necessary help and support.

Second, we would like to thank our project coordinator for providing us with various ideas and enhancements possible for uplifting of the project. Thirdly, we would like to thank our teachers for providing their knowledge and expertise on our project and helping us with feasibility study.

Similarly, we would like to express our gratitude towards our seniors who restlessly helped us upon our call regarding project aids and tips. We would also like to thank the Department of Electronics and Computer for providing us necessary study materials, technical support and financial support.

Special thanks also goes to all the colleagues, who have been assisting in the project directly as well as indirectly by reviewing and suggesting the improvements. Lastly, we would like to thank our family members for supporting us to make this project successful.

Dipesh Lamichhane (072/BEX/318)

Prabin Kumar Shrestha (072/BEX/326)

Shishir Pantha (072/BEX/339)

Shreekar Tiwari (072/BEX/340)

ABSTRACT

The Internet has transformed our lives by connecting us more easily to information and other people in the virtual world. Mobile phones then became smartphones and since then this concept has erupted and morphed into the Internet of Things, things which connect us to everyday objects. There are no end of objects that could be made “smarter”, some being more suited to this than others. Mirrors, for example, provide a large surface ideal for displaying information and interacting with. Most people have mirrors at home so the concept of a smart mirror that you can interact with is attractive and has been fantasized in many futuristic movies. Smart mirrors, such as Magic Mirror and HomeMirror have recently started to be developed by people with varying degrees of interactivity. However, so far, the features of these mirrors have been limited. This document proposes building a smart mirror from scratch using a Raspberry Pi for the hardware and custom software built on top of Raspbian, a Linux distribution.

The goal of the project is to create a Smart Mirror device that people could interact with but also to further develop the technology so that it would allow users to install and develop applications for it. The guiding principle of this endeavor is to produce a product that is simple to use, attractive to look at, and that incorporates several visual applications to make daily life more efficient.

TABLE OF CONTENTS

List of Figures

List of Abbreviations

AI	Artificial Intelligence
IoT	Internet of Things
OpenCV	Open Source Computer Vision Library
M2M	Machine to Machine
API	Application Programming Interface
SCADA	Supervisory Control and Data Acquisition
SoC	System on a Chip
OpenGL	Open Graphics Library
HOG	Histogram of Oriented Gradients
FERET	Face Recognition Technology
DIY	Do It Yourself
UI	User Interface
UART	Universal Asynchronous Receiver-Transmitter
SPI	Serial Peripheral Interface
I2C	Inter IC
LBP	Local Binary Pattern
CSS	Cascading Style Sheets
HTML	Hyperlink Text Markup Language
SBC	Single Board Chip
BSD	Berkeley Software Distribution
APT	Advanced Package Tool

1. INTRODUCTION

In an age where every individual is constantly on the move and where even a fraction of a second is worth a few dimes, there is no denying time is the king that rules us all. And there is only one rule of time, in order to be successful one must keep up with time. What if we went ahead of it though? And more importantly, how possibly can one get ahead of time? The answer is technology.

Technology has become important and inevitable part of our daily routines. With technology progressing at a rapid pace, people are also expected to be more productive and efficient in their daily activities. The use of smart phones, tablets, laptops and other similar devices has provided people with tools that help them stay productive and, more importantly, be time-efficient. However, as much as the use of such devices is time efficient, it is also time-consuming as it has become yet another task on ones' daily to-do list. In addition to this, time demands (deadlines) are most often the main cause of a person being under pressure. Therefore, good time management is the key to being on track. This does not only apply to ones' professional life only, but to their private lives as well. One activity common to us all is getting ready to leave the house in the morning for school/work duties. It also consumes a good amount of time in the morning. Checking the phone constantly to see what time it is or what the weather is like for that day does not seem much time-consuming; but it is! Being focused on the task at hand for a specific period of time is what good time management is. Therefore, a multipurpose mirror that is meant to serve as both decoration and information source can become a useful tool. With just one look at the mirror, one can have the basic information on what to wear based on weather forecast for that day or how much time they have left if they want to arrive on time at a planned destination.

Furthermore, using facial recognition to identify the user in front of the camera to give him/her specific data that they might enjoy or an event they would have to attend would provide added assistance and time management for their day.

1.1 Background Introduction

The Internet of Things (IoT) concept, is a concept which could be described as an effort to make everything smart. The goal behind this concept is to provide people with technology that would make their lives simpler and ease their daily routines. At its core, Internet of Things is about connecting devices over the Internet in a way that enables the communication between users and applications on such devices.

Likewise, face recognition systems are part of facial image processing applications and their significance in technologies have been increasing recently. They use biometric information of humans and are applicable easily instead of fingerprint, iris, signatures, etc. Face recognition systems are usually applied and preferred for people and security cameras in metropolitan life. These systems have been used for crime prevention, video surveillance, person verification, and similar security and related activities.

1.2 Motivation

As has been discussed, the primary motivation behind the project is to provide owners with a device that helps them gain time advantage compared to others. However, there are some other motivating factors that lead us to choose the project.

We live in a time when digitization is at its peak. Never have we ever been so reliant on digital devices. It is evident though, that the process of digitizing human lives does not end here. The reliance on smart devices is going to keep growing and we believe smart mirror is a part of that process of digitizing human lives for the better.

We also felt that the device could act as a virtual device. As the device is personalized, one can have access to the range of information they may desire. Calendars, important dates, news feed are some of the **widgets** we have decided to display. What is more fascinating is that, with the rapid growth in technology, smart mirrors can someday be useful to identify diseases, or may even help dress the owner. The possibilities are endless.

So, the overall motivation of the project is to help attune the owner with changing times.

1.3 Problem Definition

It is expected of every individual to keep with time due to advent of sophisticated and improved technology. We are introduced to new technologies almost every day. As

luxurious they may be or as fun the experience of using those devices are, these high-tech devices are introduced so as to assist in our work. Like the old saying *necessity is the mother of invention*, these devices are our needs rather luxuries.

The same is the case with Smart-Mirror. Though from the outset the devices is every bit a show-off, the primary purpose of the device is to save some valuable time for the owner. So often is the case that one completely forgets the most important date in his calendar. Even if s/he remembers, it would already be too late to manage anything properly. This is the problem that Smart-Mirror directly addresses. Every person in the world goes to the mirror at least once a day. Now with the system that we have developed, as soon as his face pops up in front of the mirror, more precisely in front of the camera, the mirror displays, almost magically, all the reminders and the to do lists. This, and many other problems will be a thing of the past if we have Smart-Mirrors in our homes.

Smart-mirrors can also help us in the following ways:

- It is a step forward towards digitization of human lives
- With a few tweaks, the device can act as a virtual health assistant
- The device can help boost our gaming experiences
- On addition of Kinect, the device can help us with exercising

1.4 Objective

To build a Smart Mirror with Face Recognition that shows basic information such as time, date, weather statistics, recent news headlines and information specific to the person identified in front of the mirror.

1.5 Scope and Applications

- **Smart Home**: Smart Mirror is integrated with many things connected to the smart home industry. It can be developed to control smart devices.
- **Sport**: Integrating Smart Mirror with Xbox Kinect gives us wide range of motion capabilities. Smart Mirror can help us in doing exercises and it can teach us new ones as well. Smart Mirror can be even our therapy to help us doing some needed medical exercises.

- **Shops**: Shop owners can build Smart Mirror Add-on to help customers trying new clothes, shoes or even glasses, which may be beneficial in the sense that the users are amazed with the experience, luring them to come back in the future.
- **Marketing**: The capability of Smart Mirror of showing videos can be a great idea for brand marketing. Smart Mirror can be placed in shops, homes, or public places to add great marketing features for different brands.
- **Gaming**: We can play simple games using voice command in the smart mirror. Even the complex games can be enjoyed with the users using their gestures with the use of Kinect.
- **Entertainment**: Alexa can also be integrated to the Smart Mirror, which will give it a wide range of fun capabilities like playing songs, joking, conversations and many great things.
- **Workplace**: Smart Mirror may also be applicable in offline face recognition service, which can detect, for example, the employers in and out the company and if there is stranger came into the company. Simple add-on can be added to check the employers and calculate the statistics of their attendance or fire the alarm if there is a stranger in a wrong place.

1.6 Report Organization

This report consists of the detailed study, processes involved; project requirements, project architecture, problems faced, applied solutions and screenshots related to project completion. This report begins with the basic introductory section where we discuss about the background status and problems related in our research field. Then, the same introductory part covers the objectives of our project and its possible applications and scope in our society. The next section involves literature review that covers problem definition and current measures for overcoming them. We have also discussed the possible limitations of such steps taken in the same section as well as its status in the developed societies.

The following section then involves a brief overview on requirement analysis for the project where we have discussed about the various software and language frameworks involved in our project development process. The feasibility study carried out for this project is also covered up in the very section. The following section consisting of block diagrams and

Dataflow diagrams discusses about the system architecture and processes involved in our prototype.

Then the report is focused towards the implementation details involving various unit processes involved in the system development. We discuss here about how each unit processes were accomplished stating the software, hardware and language used in the process. Moving towards the later part of the project, we have involved output and problem analysis. In this portion, we have discussed about the problems we faced before we actually ended up with final product. Then, in conclusion, the limitations of the system and possible enhancements for the project are covered up to conclude this report. The appendices involve the screenshots from our project development and output. The references section involves the sources from which we obtained materials for the completion of our work.

2. LITERATURE REVIEW

2.1 Smart Devices

A smart device, as the name suggests, is an electronic gadget that is able to connect, share and interact with its user and other smart devices. Although usually small in size, smart devices typically have the computing power of a few gigabytes. The term can also refer to a device that exhibits some properties of ubiquitous computing, including although not necessarily, artificial intelligence.

Smart devices can be characterized as follows:

- A set of system hardware & software ICT resources. This set is usually static fixed at design time
- Dynamic component-oriented resource extensions & plug-ins (Plug and play) of some hardware resources
- Remote external service access and execution
- Local, internal autonomous service execution
- Access to specific external environments: human interaction, physical world interaction and distributed ICT / virtual computing interaction.
- Ubiquitous computing properties.

Ubiquitous computing properties: Weiser's vision for ubiquitous computing can be summarized in terms of three core properties. Devices need to be networked, distributed and transparently accessible; human–computer interaction with devices is hidden to a degree from its users and devices exhibit context awareness of an environment in order to optimize their operation in that environment. Smart Devices are basically the devices that can operate to some extent autonomously and can handle a multiplicity of dynamic actions and interactions, governed by intelligent decision-making and organizational interaction. This may entail some form of artificial intelligence as well. [1]

Smart Device Environment: The term Smart Device Environment has two meanings. First, it can refer to a greater variety of device environments. Three different kinds of environments for devices can be differentiated as: virtual computing environments, physical environments and human environments. Second, the term Smart Device

Environments can also refer to the concept of a smart environment which focuses more specifically on the physical environment of the device. The physical environment is smart because it is embedded or scattered with smart devices that can sense and control part of it. [1]

2.2 IoT

Early history: The concept of the interconnected devices was first hypothesized in the 1970s, under the monikers embedded internet and pervasive computing. The primary idea was to build a network of electromechanical devices that would communicate without human intervention. [2]

Internet Appliances: It was not even a decade before the idea of self-connecting devices was realized. 1980s saw the first ever use of internet appliances. A coke machine was the front runners of such appliances. Using the web, programmers could check the status of the machine and determine the availability of cold drink without having to make a trip to the machine. It was also around this time that the idea of SCADA (Supervisory Control and Data Acquisition) was introduced. SCADA systems include hardware and software components. The hardware gathers and feeds data into a computer that has SCADA software installed, where it is then processed and presented it in a timely manner. The evolution of SCADA is such that late-generation SCADA systems developed into first-generation IoT systems. [2]

Internet of Things: The word Internet of Things was first coined by Kevin Ashton, co-founder of the Auto-ID Center at MIT, in a presentation he made to Procter & Gamble (P&G) in 1999. An MIT professor by the name of Neil Gershenfeld also gave an indication as to where the IoT was headed to in his book, *When Things Start to Think*. Ever since, the IoT has only found more applications in human lives. [2]

IoT basically started from M2M to being a sensor network of billions of smart devices, which has led many to use the word IoT ecosystem, so as to represent a huge network of interconnected devices. [3]

2.3 Face Detection

The idea of detecting faces in a picture is not completely novel and yet, face detecting algorithms are some of the newest in the field of computer science. However, it is not due

to lack of trying that the world got acclimatized to face detection only recently. In fact efforts have been put throughout different phases of time and while the algorithms have been long time in the coming, there is no denying that we finally have reliable face detecting algorithms.

Finding faces in images with controlled background: Images are used in plain mono color backgrounds and the faces are detected by recognizing the face boundaries. Though simple in its application, the method has very limited use.

Finding faces by color: The color of the skin is the reference here. However, the primary disadvantage is that it doesn't work with all kinds of skin colors, and is not very robust under varying lighting conditions

Finding faces by motion: This method was contemplated for the real time video. Here, calculating only the moving area suffices in this method. As for an example, the method would quickly detect the presence of face in the frame, if the human eye blinked. The method, though very appealing, is a bit incomplete in the sense that it relies on changes in the face, and any insignificant movement would not be take into account. The other aspect which the method failed to completely address was the situation of objects moving in the background, which it would wrongly recognize as a face.

Using a mixture of the above: Combining several good approaches normally yields an even better result.

Finding faces in unconstrained scenes: This approach addresses most of the issues.

Model-based Face Tracking: This is an edge-based method, using geometric models. Two top performing methods were published in the early 2000s:

- **Real-Time Face Detection Using Edge-Orientation Matching:** In this method, a person's image is mathematically modeled by a set of 2-D equations.
- **Robust Face Detection Using the Hausdorff Distance:** In this method, the face of the person is approximated to various geometrical shapes.

Weak classifier cascades

- **Robust Real-Time Face Detection:** The breakthrough in face detection happened with Viola & Jones. Using a cascade of "weak-classifiers", using simple Haar features, can yield impressive results. This approach is now the most commonly used algorithm for face detection. A basic implementation is included in OpenCV.

- **HOGs and Deep Learning:** Deep Learning (using multi-layered Neural Networks), especially for face recognition more than for face finding, and HOGs (Histogram of Oriented Gradients) are the current state of the art (2017) for a complete facial recognition process.

2.4 Face Recognition

Face recognition technology is used excessively used in the modern world. From regular attendance in institutions to matching criminals' faces to even improving securities in airports, facial recognition systems are as common as any modern technology. However, the concept of face recognition is not new as one may perceive, which has a history that dates back to as early as 1960s. [4]

Here are some key events in the history of facial recognition:

Manual Measurements by Bledsoe (1960s): Woodrow Wilson Bledsoe, often credited to as the father of facial recognition, developed a system that could classify photos of faces. In this method, people used to input horizontal and vertical coordinates on a grid using a stylus that would emit electromagnetic pulses. This system used to manually record coordinate locations of various facial features. Recorded in a database, the system was given a new photograph of an individual, which the system would be able to retrieve from the database that most closely resembled the person's image. However, due to lack of computational advantage, it was very limited in its application.

Increased Accuracy with 21 Facial Markers (1970s): In the 1970s, Goldstein, Harmon, and Lesk were able to add increased accuracy to a manual facial recognition system. They used 21 specific subjective markers including lip thickness and hair color in order to identify faces automatically. As with Bledsoe's system, the actual biometrics had to still be manually computed.

Eigenfaces (Late 1980s-Early 1990s): In 1988, Sirovich and Kirby began applying linear algebra to the problem of facial recognition. What became known as the Eigenface approach started as a search for a low-dimensional representation of facial images. Sirovich and Kriby were able to show that feature analysis on a collection of facial images could form a set of basic features. They were also able to show that less than one hundred values were required in order to accurately code a normalized face image. [5]

In 1991, Turk and Pentland expanded upon the Eigenface approach by discovering how to detect faces within images. This led to the first instances of automatic face recognition. Their approach was constrained by technological and environmental factors, but it was a significant breakthrough in proving the feasibility of automatic facial recognition.

FERET Program (1993-2000s): The Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology rolled out the Face Recognition Technology (FERET) program beginning in the 1990s in order to encourage the commercial face recognition market. The project involved creating a database of facial images. The database was updated in 2003 to include high-resolution 24-bit color versions of images. Included in the test set were 2,413 still facial images representing 856 people. The hope was that a large database of test images for facial recognition would be able to inspire innovation that might result in more powerful facial recognition technology. [5]

Face Recognition Vendor Tests (2000s): The National Institute of Standards and Technology (NIST) began Face Recognition Vendor Tests (FRVT) in the early 2000s. Building on FERET, FRVTs were designed to provide independent government evaluations of facial recognition systems that were commercially available, as well as prototype technologies. These evaluations were designed to provide law enforcement agencies and the U.S. government with information necessary to determine the best ways to deploy facial recognition technology. [5]

Law Enforcement Forensic Database (2009): In 2009, the Pinellas County Sherriff's Office created a forensic database that allowed officers to tap into the photo archives of the state's Department of Highway Safety and Motor Vehicles (DHSMV). By 2011, about 170 deputies had been outfitted with cameras that let them take pictures of suspects that could be cross-checked against the database. This resulted in more arrests and criminal investigations than would have otherwise been possible.

Social Media (2010-Present): Beginning in 2010, Facebook began implementing facial recognition functionality that helped identify people whose faces may be featured in the photos that Facebook users update daily. While the feature was instantly controversial with the news media, sparking a slew of privacy-related articles, Facebook users at large did not seem to mind. Having no apparent negative impact on the website's usage or popularity, more than 350 million photos are uploaded and tagged using face recognition each day.

Face Recognition "Inevitable" For Retail (2017): As face recognition is adopted by retail faster than any other industry, experts are taking note. In a recent webinar, D&D Daily Publisher and Editor Gus Downing stated that face recognition is on an "inevitable path to retail adoption." Downing, considered one of the foremost loss prevention thought leaders, is just one expert that now sees massive advantages for retailers who use a face recognition system.

Iphone X (2017): Apple released the iPhone X in 2017, advertising face recognition as one of its primary new features. The face recognition system in the phone is used for device security. The new model of iPhone sold out almost instantly, proving that consumers now accept facial recognition as the new gold standard for security.

2.5 Smart Mirror

A Smart mirror displays information required to its owner while also serving the usual reflection purpose. The magic mirror, which could also be termed as its twin has had increased application so far and plenty of research has also been done.

First Generation Magic Mirrors (2007): The first generation magic mirrors were primarily used for advertising purposes. A screen would display an advertisement of a product of a band. As anyone would approach the screen, it would transform into a mirror and the advertisement would be reduced to one corner of the screen, in the hope of creating curiosity to the observers. Adidas introduced an interactive wall in the same year, which allowed the customers to scroll through its products. [6]

Second Generation Magic Mirrors (2008): Apple introduced a Photo Booth that would initiate the second generation of magic mirrors. The photo booth allowed people to add thumbnails and add effects to the image already captured. Photo filter functions were introduced in 2011 in the mirrors that allowed to add effects and change various attributes of the photo. In the same year, slimming algorithm was also added that allowed the trimming of one's body size in a proportionate and well-rounded manner. Outside of the photo editing, they were also used for recreational purposes like in a catch-and-win game.

The Cybertecture Mirror, developed by James Law, displays basic information such as date, time, and weather. It also allows social network integration, so the user has access to their social accounts even when in bathroom. However, the price of the mirror does not make it

quite affordable to everyone because the cost of customization could go from 3,600 to 7,700 dollars. [6]

Third Generation Magic Mirrors (2011): These mirrors were Kinect integrated. A Kinect is a motion sensing input device that was primarily used for gaming purposes that senses the movements of the players and the motions are mimicked in the screen. The use of the Kinect allowed for various exciting prospects like virtual dressing apps that allowed for trying on the dresses without having to take the clothes off. Gesture recognition and coupon vending machines also gained considerable popularity. [6]

In 2012, at the Consumer Electronics Show held in Las Vegas, Samsung presented Smart Window: a window that is also a device. Although not a mirror, this device allows users to access applications such as weather or social apps all the while serving as a window.

Fourth Generation Magic Mirrors (2013): The magic mirrors now started coming with a control panel, and fourth generation was ushered in with the advent of Control Panel version 1.0. Various interactive advert apps were already popular by 2013. The eyewear trying app was incredibly popular as brands across the world opted to allow the customers to try on many different glasses before choosing the desired glass.

Control Panel version 2.0 was also introduced in 2014 which was used by a hair style studio, which would allow the hairstylist to try on a few hair colour until the customer would be satisfied. The Memomi mirror has been used as a replacement for dressing rooms in shops that sell clothes. The mirror gives them a 360-degree-view of their outfits as well as the possibility to modify their looks by changing the colour or adding accessories to the outfit.

With the introduction of Control Panel version 3.0, the magic signages are set to revolutionize the fashion industries, while also making it unbelievably simple for the customers to interact with the web, as it allows uploading, managing and storing of the viewers' attitude towards the product. Instaprint (2015) is an example of magic signage.

Aside from the above mentioned products, there are also smaller-scale projects that involved piecing components together to achieve the functionality of a smart mirror. These are mostly done as DIY projects and are used for individual purposes. Max Braun’s bathroom mirror stands out from this group (DIY projects).

3. REQUIREMENT ANALYSIS

3.1 Hardware Requirements

a. Two way mirror: A two way mirror is a mirror that allows the passage of light through it, while also maintaining its reflective property. The glass is coated with a thin and almost transparent layer of metal (usually aluminium). The idea here is to make one side dark by 6 times (ideally) than the other.

A glass two way mirror is 70% reflective and 11% transparent**.**

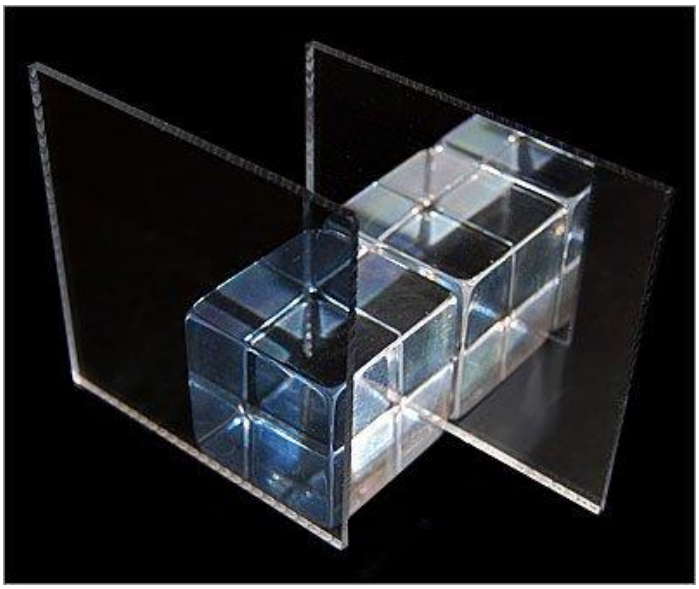


Figure 3-1: Two Way Mirror

[7]

b. LED Monitor: The monitor is where we display the UI webpage. For this project, we have used a 19” LED monitor.

c. Raspberry Pi: Depending on the model, the Raspberry Pi has either 40 or 26 dedicated interface pins. In all cases, these include a UART, an I2C bus, a SPI bus with two chip selects, I2S audio, 3V3, 5V, and ground. [8] The maximum number of GPIOs can theoretically be indefinitely expanded by making use of the I2C or SPI bus.

There is also a dedicated CSI-2 camera port for the Raspberry Pi Camera Module, and a DSI display port for the Raspberry Pi LCD touchscreen display.

Figure 3-2: Raspberry Pi

All Pi models up to the Raspberry Pi 3 have a GPU that provides OpenGL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile encode and decode.

The reason we opted to choose Raspberry Pi over other SBCs, is because, for its price, we found it to be the most efficient of all other devices. It is the most reliable device and the only other SBC that came close to its specifications was Orange Pi Prime. It is even cheaper than the device we have used and although it matches Raspberry Pi 3B+ in terms of specifications, the device has been deemed largely useless, due to its unreliability as the software is found to be buggy. Other options such as Banana Pi and Odroid series were left out of consideration either due to their unavailability in the accessible market or their extremely high prices.

Also, as compared to its predecessors in the raspberry pi series, the 3B+ model is the latest one, and has higher processing and Ethernet speed. Hence, we opted for the Raspberry Pi 3B+.

d. Pi Camera Module: The Camera Module is a small PCB that connects to the CSI-2 camera port on the Raspberry Pi using a short ribbon cable. It provides connectivity for a camera capable of capturing still images or video recordings.

Figure 3-3:Pi Camera

The Camera Module connects to the Image System Pipeline (ISP) in the Raspberry Pi's SoC, where the incoming camera data is processed and eventually converted to an image or video on the SD card (or other storage). It's the only camera that is compatible with the Raspberry Pi. The Camera Module V2 is capable of taking photos up to 8 megapixels (8MP). It supports 1080p30, 720p60 and VGA90 video modes [9], as well as still capture. The original Camera Module is capable of taking photos up to 5 megapixels and can record video at resolutions up to 1080p30. [9]

Compared to the 720p of the webcam of the PCs in hand, this module was a really a step above. The webcam also does not treat backlighting as well as the pi camera and the edges of objects were ragged. Also, the fact that the camera app was slow for the webcam, made the pi camera module a step above the PC webcam.

3.2 Software Requirements

a. Operating System:

Since we opted to use Raspberry Pi as the computer, there was no choice other than to use Linux Operating System, since Pi is especially built for Linux. However, this was more a blessing in disguise for the project as it turned out to be way cheaper than other Operating Systems. Also, it is advantageous due to its incorruptibility and its unmatched reliability, on top of being a very stable operating system.

Linux is a family of open source Unix-like operating systems based on the Linux kernel and is typically packaged in a Linux distribution include the Linux kernel and supporting system software and libraries, many of which are provided by the GNU Project.

We preferred to use **Ubuntu** among the Linux distributions as it was found to be easiest to work with. Admittedly, our choice of the Linux distribution was also aided by its popularity and its convenience, especially for the beginners.

Ubuntu which translates as "humanity to others" in African philosophy, is a free and open-source Linux distribution based on Debian. Ubuntu is officially released in three editions: Desktop, Server, and Core (for IoT devices and robots) and the default file manager is GNOME Files.

Ubuntu operates under the GNU General Public License (GPL) and all of the application software installed by default is free software. User programs run with low privileges and cannot corrupt the operating system or other users' files. For increased security, the sudo tool is used to assign temporary privileges for performing administrative tasks, which allows the root account to remain locked and helps prevent inexperienced users from inadvertently making catastrophic system changes or opening security holes.

The system requirements vary among Ubuntu products. For the Ubuntu desktop release 16.04 LTS, a PC with at least 2 GHz dual-core processor, 2 GB of RAM and 25 GB of free disk space is recommended. For less powerful computers, there are other Ubuntu distributions such as Lubuntu and Xubuntu. Ubuntu also supports the ARM architecture.

b. Python

Python is a general-purpose programming language that can be used on any modern computer operating system. Python is an interpreted, high-level, general-purpose

programming language. It can be used for processing text, numbers, images, and scientific data and just about anything else you might save on a computer. Since Python is an interpreted language, it is not converted to computer-readable code before the program is run, but at runtime.

It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. The standard library has two modules (itertools and functools) that implement functional tools.

c. PHP

PHP (Hypertext Preprocessor) is general purpose programming language, created by Rasmus Lerdorf in 1994. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems (CMS), and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. PHP has a direct module interface called Server Application Programming Interface (SAPI), which is supported by many web servers including Apache HTTP Server, Microsoft IIS, Netscape (now defunct) and iPlanet. Main Application of PHP is Web Application making web pages dynamic.

Frameworks for PHP are Laravel, Codeigniter, Symfony, CakePHP, Yii, Zend, Phalcon, FuelPHP, etc.

In this project, CodeIgniter (CI) is use as PHP framework for web part. Phpmyadmin Mysql has been used for database.

CodeIgniter is an Application Development Framework,for people who build web sites using PHP which uses OOP. Its goal is to develop projects much faster than that could be wrote from scratch, by providing a rich set of libraries for commonly needed tasks, as well

as a simple interface and logical structure to access these libraries. CodeIgniter lets creatively focus on the project by minimizing the amount of code needed for a given task.

In this particular project, CI was used by configuring in HMVC (High Model View Controller), by default is MVC. Here, frontend surface for mirror is implemented in MVC and backend system is implemented in HMVC under same CodeIgniter System making project more structured under directory and easy to work on. In HMVC, controllers and models work as it is, but we can create module which is simple directory for each task of the project which further have its own controller, model and views.

When url is hit, which contains controller or manual route to controller, then controller loads the particular view file. All functional things happens in controller, with database work on model and view file has design part, which is simple working principle of MVC.

d. Libraries

OpenCV: (Open Source Computer Vision Library) is a BSD-licensed open source computer vision and machine learning software library, mainly aimed at real-time computer vision. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS, although it is primarily written in C++. OpenCV leans mostly towards real-time vision applications.

NumPy: It is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. The NumPy array such as universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

At the core of the NumPy package, is the *ndarray* object. This encapsulates *n*-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. NumPy fully supports an object-oriented approach, starting, once again, with *ndarray*.

There are several important differences between NumPy arrays and the standard Python sequences. NumPy arrays have a fixed size at creation, unlike Python lists which can even grow dynamically. Also, the elements in a NumPy array ought to be of the same data type, and thus will be the same size in memory.

NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

However, with all the pros, there are cons of using Numpy as well. Inserting or appending entries to an array is not as trivially possible as it is with Python's lists. Also, algorithms that are not expressible as a vectorized operation will typically run slowly because they must be implemented in pure Python, while vectorization may increase memory complexity of some operations from constant to linear, because temporary arrays must be created that are as large as the inputs.

e. IDE:

Visual Studio Code: It is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.

Visual Studio Code can be extended via plug-ins, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, debuggers, perform static code analysis, add code linters, using the Language Server Protocol and connect to additional services.

Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

Visual Studio Code has out-of-the-box support for almost every major programming language. Several are included by default, for example, JavaScript, CSS, and HTML while other language extensions can be found and downloaded for free from the VS Code Marketplace.

Visual Studio Code is widely reviewed to be fast and lightweight, and is considered to be flexible across various domains such as Java, JavaScript, Go, Node.js and even C++.

PyCharm: PyCharm is a cross-platform IDE (with windows, macOS and Linux versions) used in computer programming, specifically for the Python language.

PyCharm's huge collection of tools out of the box includes an integrated debugger and test runner, Python profiler, a built-in terminal, integration with major VCS and built-in database tools remote development capabilities with remote interpreters, an integrated ssh terminal and integration with Docker and Vagrant.

PyCharm's smart code editor provides first-class support for Python, JavaScript, CSS, popular template languages and more. It has also made it easier for users to jump to any class, file or symbol any other IDE action or tool window with smart search.

PyCharm provides API so that developers can write their own plugins to extend PyCharm features.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Block Diagram

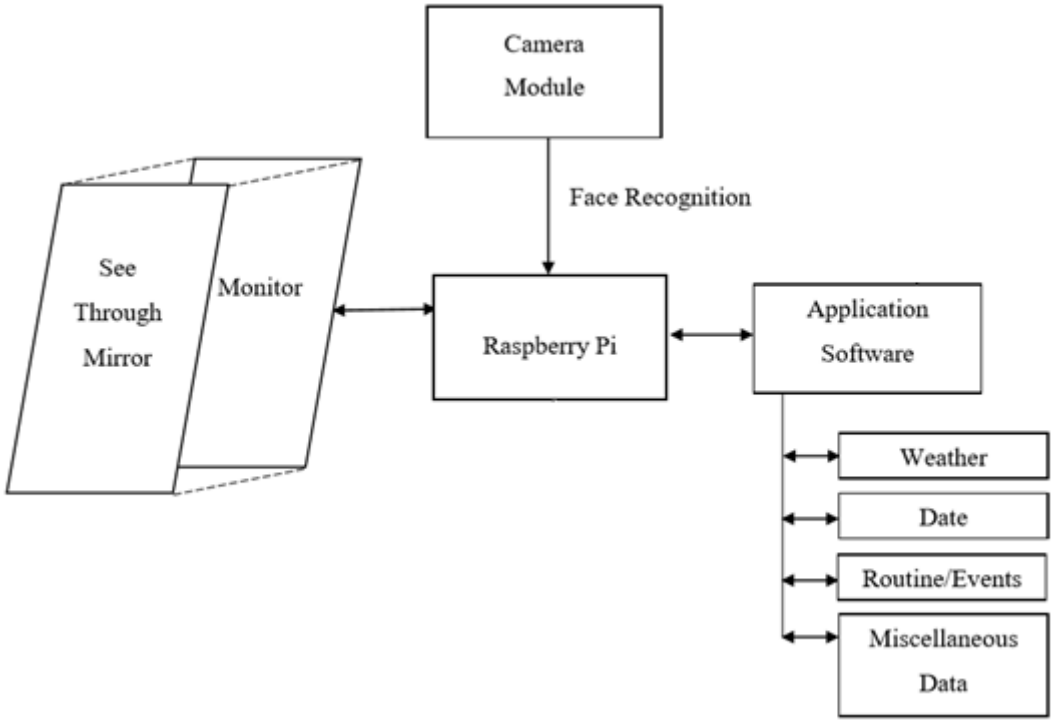


Figure 4-1: System Block Diagram

4.2 Description of Working Principle of Face Detection and Face Recognition

The first part deals with designing the box that is used as a container for the monitor, controller board, and mirror. Both monitor and Raspberry Pi are placed inside the box and secured to avoid any possible damage to any of the components while moving the box around. Once these two components are fixed, the mirror will cover the front of the box. Any information that is to be displayed on the mirror will actually be displayed on the monitor. The characteristics of the mirror are such that the front side is reflective and acts as a mirror, while the back side is transparent, so anything that is displayed on the monitor can be seen on the mirror surface. Furthermore, to avoid any possible movements of the monitor and controller board that could cause damage to these components, the box will need compartments for them to ensure they stay in place.

For the development stage, an application will be developed that generates the specified information that include time, date, weather forecast, and list of tasks. The background of the application will be black and the font color will be white to ensure maximum reflection

of the mirror. The application implements a sign-up/login type feature using face recognition, so that the user has their own 'account' and is able to customize the mirror and related information they might need in the future. Upon successful recognition of a preprogrammed face, they will be redirected to the page where the relevant information is stored. The Raspberry Pi is configured in such a way that, when it boots, it automatically launches the specified application.

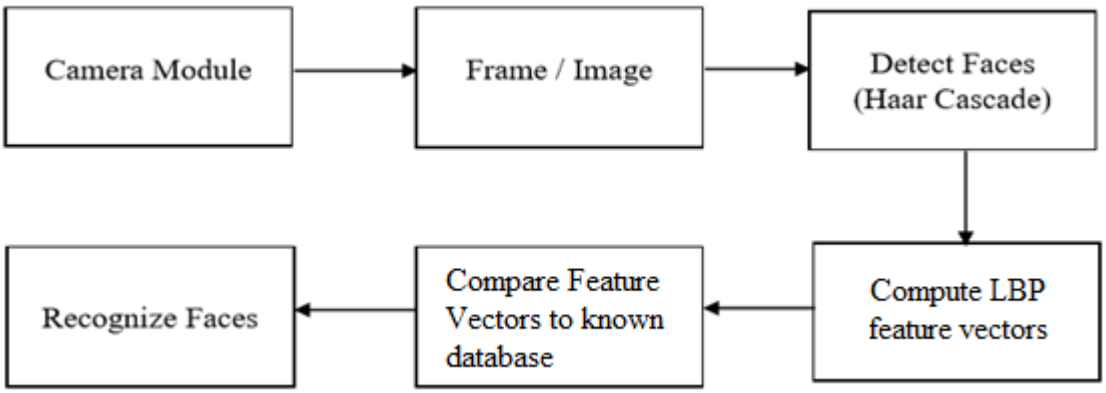


Figure 4-2: Facial Detection and Recognition

For the facial recognition, a dataset of example images containing faces to be recognized are first gathered. Then, deep neural network is used to compute a 128-d vector (i.e., a list of 128 floating point values) [10] that will quantify each face in the dataset. OpenCV's Haar cascade can then be used as a Python script to detect and localize the face. The actual face recognition is, therefore, carried out using OpenCV, Python, and deep learning.

4.2.1 Haar Cascade

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video. 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. [10]

The algorithm has four stages:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost Training
4. Cascading Classifiers

It is well known for being able to detect faces and body parts in an image, but can be trained to identify almost any object.

For face detection, initially, the algorithm needs a lot of positive images of faces and negative images without faces to train the classifier. Then we need to extract features from it. [11]

First step is to collect the Haar Features. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. [10]

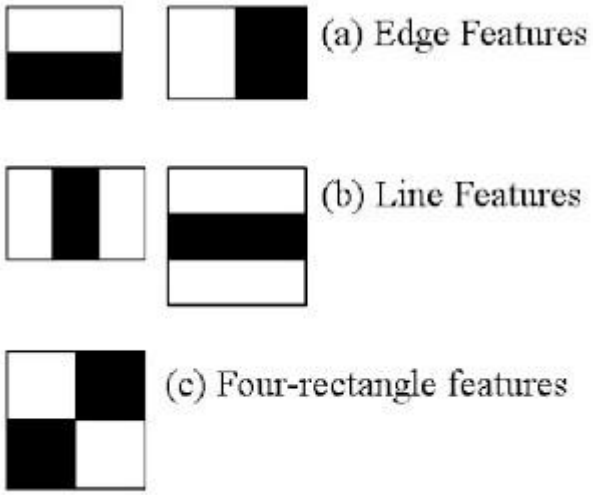


Figure 4-3: Haar Cascade Features

But among all the features that are calculated, most of them are irrelevant. Top row may show two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant.

Adaboost comes to the rescue here as it selects the best features and trains the classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

The process is as follows:

During the detection phase, a window of the target size is moved over the input image, and for each subsection of the image and Haar features are calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because

each Haar feature is only a "weak classifier" (its detection quality is slightly better than random guessing) a large number of Haar features are necessary to describe an object with sufficient accuracy and are therefore organized into cascade classifiers to form a strong classifier.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. *Positive* indicates that an object was found and *negative* indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

The stages are designed to reject negative samples as fast as possible. The assumption is that the vast majority of windows do not contain the object of interest. Conversely, true positives are rare and worth taking the time to verify.

- A *true positive* occurs when a positive sample is correctly classified. [11] [12]
- A *false positive* occurs when a negative sample is mistakenly classified as positive. [11] [12]
- A *false negative* occurs when a positive sample is mistakenly classified as negative. [11] [12]

To work well, each stage in the cascade must have a low false negative rate. If a stage incorrectly labels an object as negative, the classification stops, and you cannot correct the mistake. However, each stage can have a high false positive rate. Even if the detector incorrectly labels a non-object as positive, you can correct the mistake in subsequent stages. Adding more stages reduces the overall false positive rate, but it also reduces the overall true positive rate.

Cascade classifier training requires a set of positive samples and a set of negative images. One must provide a set of positive images with regions of interest specified to be used as positive samples. The Image Labeler may be used to label objects of interest with bounding boxes. The Image Labeler outputs a table to use for positive samples.

4.2.2 LBP

The original LBP operator labels the pixels of an image with decimal numbers, called Local Binary Patterns or LBP codes, which encode the local structure around each pixel. Each pixel is compared with its eight neighbors in a 3x3 neighborhood by subtracting the center pixel value; the resulting strictly negative values are encoded with 0 and the others with 1. A binary number is obtained by concatenating all these binary codes in a clockwise direction starting from the top-left one and its corresponding decimal value is used for labeling. The derived binary numbers are referred to as Local Binary Patterns or LBP codes.

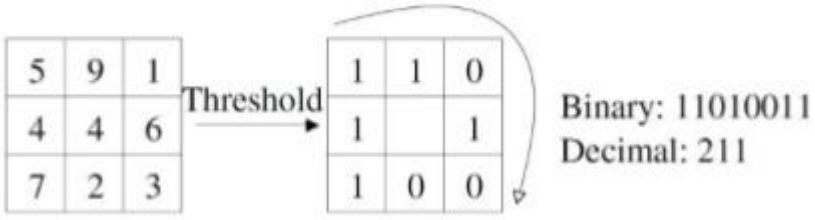


Figure 4-4: LBP Algorithm

One limitation of the basic LBP operator is that its small 3x3 neighborhood cannot capture dominant features with large scale structures. To deal with the texture at different scales, the operator was later generalized to use neighborhoods of different sizes. A local neighborhood is defined as a set of sampling points evenly spaced on a circle which is centered at the pixel to be labeled, and the sampling points that do not fall within the pixels are interpolated using bilinear interpolation, thus allowing for any radius and any number of sampling points in the neighborhood.

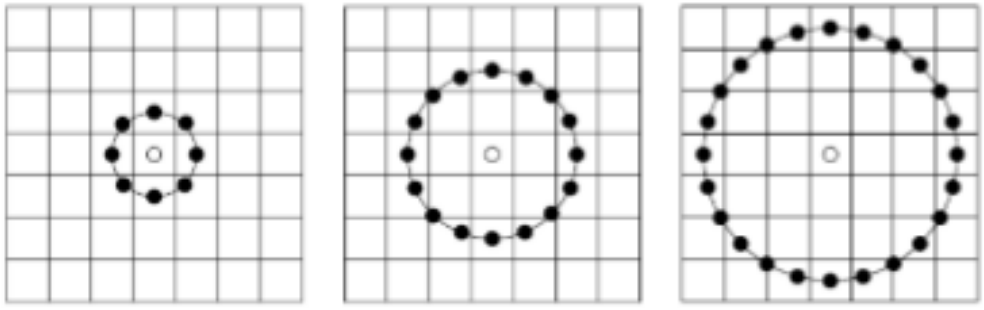


Figure 4-5: Improved LBP Algorithm

Formally, given a pixel at (x_c, y_c) the resulting LBP can be expressed in decimal form as:

$$LBP_{P,R}(x_c, y_c)=\sum_{P=0}^{P-1} s(i_P-i_C)2^P \ldots\ldots\ldots\ldots\ldots\ldots\ldots (1) \quad [13]$$

where i_C and i_P are respectively gray-level values of the central pixel and P, the surrounding pixels in the circle neighborhood with a radius R, and function $s(x)$ is defined as:

$$s(x)=\begin{cases}1 \; if \; x \geq 0 \\ 0 \; if \; x < 0\end{cases} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2) \quad [13]$$

By the definition above, the basic LBP operator is invariant to monotonic gray-scale transformations preserving pixel intensity order in the local neighborhoods. The histogram of LBP labels calculated over a region can be exploited as a texture descriptor. The operator $LBP_{P, R}$ produces 2^P different output values, corresponding to 2^P different binary patterns formed by P pixels in the neighborhood. If the image is rotated, these surrounding pixels in each neighborhood will move correspondingly along the perimeter of the circle, resulting in a different LBP value, except patterns with only 1s and 0s. In order to remove rotation effect, a rotation-invariant LBP is proposed in

$$LBP^{ri}{}_{P,R}=min\{LBP_{P,R,}\, i|i=0,1,2,\ldots,P-1\} \ldots\ldots\ldots\ldots\ldots\ldots (3) \quad [13]$$

where ROR(x, i) performs a circular bit-wise right shift on the P-bit number x i times.

The $LBP^{ri}{}_{(P, R)}$ operator quantifies occurrence statistics of individual rotation invariant patterns corresponding to certain micro-features in the image; hence, the patterns can be considered as a feature detector. However, in [13], it was shown that such a rotation-invariant LBP operator does not necessarily provide discriminative information, since the occurrence frequencies of the individual patterns incorporated in $LBP^{ri}{}_{P, R}$ vary greatly and the crude quantization of the angular spaces at 45° intervals. It has been shown that certain patterns contain more information than others. It is possible to use only a subset of 2^P binary patterns to describe the texture of images. Ojala et al. named these patterns uniform patterns, denoted $LBP^{U2}{}_{(P, R)}$.

A local binary pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the corresponding bit string is considered circular. For instance, 00000000 (0 transitions) and 01110000 (2 transitions) are both uniform whereas 11001001 (4 transitions) and 01010011 (6 transitions) are not. It is observed that the uniform patterns

account for around 90% of all the patterns in a (8, 1) neighborhood and around 70% in a (16, 2) neighborhood in texture images. [13]

A similar experiment was conducted on the FERET database, and it was found that 90.6% of the patterns in a (8, 1) neighborhood and 85.2% in a (8, 2) neighborhood are uniform. More recently, Shan and Gritti verified validity of uniform patterns for representing faces from the viewpoint of machine learning. Specifically, they applied AdaBoost to select the discriminative patterns for facial expression recognition, and their experiments demonstrated that, using LBP (8, 2) operator, 91.1% of these selected patterns are uniform. Accumulating the non-uniform patterns into a single bin yields an LBP operator with less than 2p labels. For example, the number of labels with the neighborhood of 8 pixels is 256 for the standard LBP but only 59 for LBP^{U2}. [13]

5. IMPLEMENTATION DETAILS

5.1 Design and Layout

5.1.1 Casing Design

The LED monitor that we have used is 19” long diagonally with its sides having dimensions of18”x11.5”. Hence, an 18”x15” glass block was constructed, whose thickness was 4mm. Due to the unavailability of the two way mirror in accessible markets, we compromised for a two way sticker. The sticker was glued on one of the glass surfaces. The reflections and transparencies of the sticker-mirror combination was found good enough to carry on with our project.

A casing was built to hold the monitor in its place, and also to keep the hardware devices safely inside of it. The pi camera was placed

5.1.2 Webpage Layout

The UI webpage was designed with the help of HTML, CSS and Javascript. The page basically contains all the personalized information, as per the request of the user. The information we have displayed in the webpage include:

- **Time and Date:**

On the top left corner of the webpage, we can see the time and date of that day. This was created with the help of Javascript, and were dynamically evaluated.

- **Name of User:**

The name of the user is shown to be centrally aligned in the top part of the webpage, after the word Hello. The name was extracted from the database.

- **To-do List:**

To-do-list that were added by detected user are displayed. To-do-list with assigned date in the same day or following days are only shown in ascending order. As much as eight activities can be displayed in the webpage and the activities whose dates are past are not displayed on the screen. The to-do-list is placed towards the left of the screen

- **Headlines:**

The user id from the database also provides with the preferred online news source. If the user hasn't opted for any newspaper, the headlines of the default news source will be displayed. The headlines are extracted from the RSS of newspapers. All newspapers have Really Simple Syndication (RSS) feed. They are obtained from their particular RSS feed URL.

RSS is a type of web feed which allows users and applications to access updates to websites in a standardized, computer-readable format. These feeds can, for example, allow a user to keep track of many different websites in a single news aggregator. The news aggregator will automatically check the RSS feed for new content, allowing the content to be automatically passed from website to website or from website to user. This passing of content is called web syndication. Websites usually use RSS feeds to publish frequently updated information, such as blogentries, news headlines, or episodes of audio and video series. A standard XML file format ensures compatibility with many different machines/programs. RSS feeds also benefit users who want to receive timely updates from favourite websites or to aggregate data from many sites.

Given URL of online newspaper feed is in XML format, has to be converted into array-object format where XML tags becomes array key, and data are extracted accordingly. These data are then combined into multidimensional array with each having title, detail and date. These headlines title are shown in mirror in descending order of time created using loop.

Weather: The maximum and minimum temperature of the current day and daily basis weather up to eight days are displayed in the webpage. Both includes SVG images for better UI. Weather Feature has two phases to show get weather data.

1. Geolocation API using IP Address

In PHP, IP address is extracted, and using this IP and ipinfo.io API, Geolocation is obtained. HTTP request (https://ipinfo.io/<ip-address>?token=<API Key>) where <ip-address> is IP obtained and <API Key> is API key obtained from the system, returns JavaScript Object Notation (JSON) which has many attributes like network information, ISP and geolocation. This geolocation obtained is in format latitude,longitude.

2. Weather API From Geolocation

After obtaining latitude and longitude, using these parameters, and calling weather API (in this case Darksky) returns all detail about weather in particular locaiton. HTTP request (https://api.darksky.net/forecast/<appID>/<lattitude,longitude>?units=si) where <latitude,longitude> is geolocation obtained in phase-i and <appID> is API key obtained from the system, returns (JSON) which has many information with SI units.

Information obtained are hourly weather reports, daily weather reports, weather alerts (which are only allowed by some geolocation), and weather reports involves temperature, relative temperature, humidity, sunrise time, icons, summary, UNIX time, etc. Among these reports, current weather and daily weather information are only displayed in mirror.

JSON obtained is JSON decoded in php which then is in array and object format. Required data are taken accordingly.

The Design implements using JS:

- Headlines and To-list are shown in descending order along decreasing opacity. With the help of JS, their opacity were made decrease gradually.
- Using JS, daily weather were displayed as slider. They were made vanishing and new data comes with gradually increasing opacity, displayed in block and then again made vanished gradually and cycle continues.

5.2 Programming

5.2.1 Face Detection

As already mentioned, we have used Haar features for the face detection. The idea is to make use of the difference of light intensities between two surfaces along the face. An example would be the differing light intensities between forehead and eyes and then in the cheek areas right below the eyes. Another example is between nose and cheek parts, where nose is brighter compared to its edges.

The image is first converted to greyscale images, because of the added advantage of having to deal with just one color. Here, the only varying attribute between parts of faces would be light intensities and their differences.

Various functions need to be imported from OpenCV, which include:

cv2.imshow: This is a cv2 function used to display the image. It also takes two arguments: the first one is the name of the window that will pop-up to show the picture and the second one is the image to be displayed.

cv2.cvtColor: It is an OpenCV function to convert images to different color spaces. It takes as input an image to transform, and a color space code and returns the processed image.

cv2.CascadeClassifier: It takes as input the training file of the classifier we want to load and loads it for us. **detectMultiScale** is a function which takes three input parameters, one of which is the greyscale image. If there is a face it returns a list of face positions in the form of co-ordinates.

5.2.2 Face Recognition:

The face thus detected, represented in the form of co-ordinates, is re-ordered and then we quantify the face with the help of 128-d encodings. The values are then matched with the stored images and the values that most resemble the matches, the corresponding individual's name is displayed on top of the square (which identifies the detection of faces).

In case of multiple matches, the name of person with more matches is displayed on the screen. The similarity, or the difference between the images and individuals can be compared with the help of histogram

A **Histogram** is created by plotting pixel percentage values against the pixel values (generated using the LBP algorithm) of a particular area. As such, we will have a plethora of such areas, as each area is only a few pixels. Hence, it may be a slower process, but it is accurate representation of the human face.

5.2.3 Backend web development:

Backend was a crucial part of the system we have designed. It allows the user to change his personalized and preferred information, while also giving a platform for webpage layout. The following are the key elements of our backend part.

- **User System:** For User System, username and password are used to log into the system. Here, username is unique, whose uniqueness is validated through code and using database unique property. Passwords are encrypted by md5 password and saved to database, making it secured. In the Login System, confirmation password feature is also implemented. User system has two user roles, Admin user and Normal user. Admin

Users are accessible to all tasks in the project while Normal user are not accessed to User Management System.

Further, it includes data like email, full name, status to check whether the user is active or not, created date of user, last logout date when user is logged out of system. User can be deleted only by Admin Users.

- **Dashboard System:** It is the system when user logs in. It has links to different system like task management, newspaper selection, password change and user management system. It cannot be accessed when user does not log in i.e., when user log in session is not created but will be redirected to login system.

- **Newspaper System:** In this system, different online newspapers can be selected. These newspapers are stored in a single data-table. Different pivot table is created to store particular newspaper selected by particular user. Headline News of selected newspaper are shown in smart mirror platform.

- **To-do Management System:** This system is to store to-do tasks. This includes to-do task title, priority like urgent, medium and normal, to-do assign date, details and publish status whether to publish to-do task or not. In to-do date selection library is used for better User Interface (UI). Feature to add, edit and delete are also the part of this system. In detail input field, JavaScript (JS) ckediter library is used for better writing features like font style, size, color, etc. These to-list are list in smartmirror screen. Upcoming to-do-list and today to-do lists can be viewed separately.

- **Password Change:** Password change system allows user to change their password, where user has to input old and two new password, one as confirmation password. All list involves feature of search for easiness in order to search in case of long list, edit and delete system. Design in backend uses bootstrap. Pagination was also added in backend list in case of long list.

6. RESULTS AND ANALYSIS

6.1 Face Detection

Upon showing our faces to the camera, the following is what we see on display.

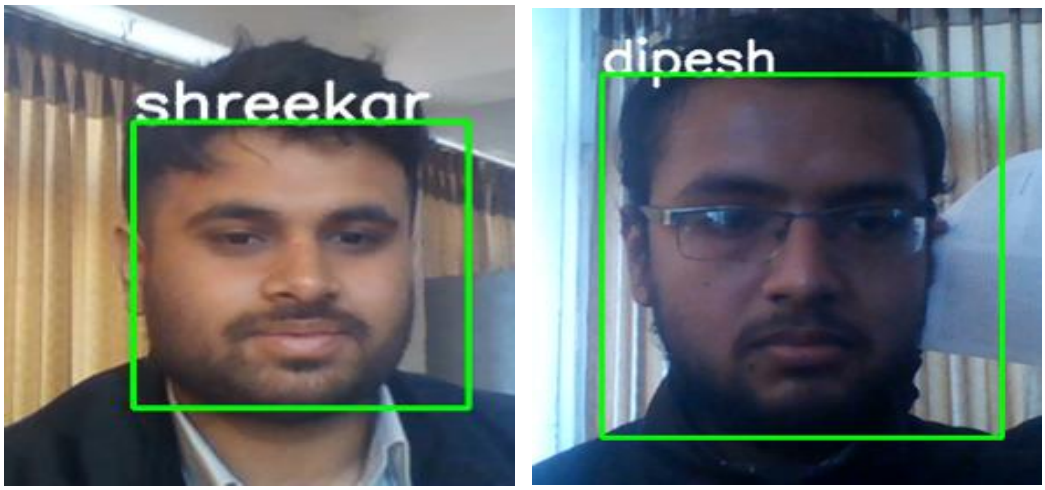


Figure 6-1: Face Detection

As a face is detected, a square "encloses" the face. The name on top of the square is the name of the individual whose face resembles with the values obtained after image processing.

There were instances when names were flipping from one name to another. This is due to the fact that the image that camera has scanned is similar to both the images. The one with most matches was eventually chosen by the system. Obviously, no such squares were shown when face was not detected, and when an "unfamiliar" face showed up against the camera, there was just a square around the face, indicating the presence of a face. However, no name propped up in such cases.

6.2 Face Recognition

Figure 6-2: LBP Representation of Face Image

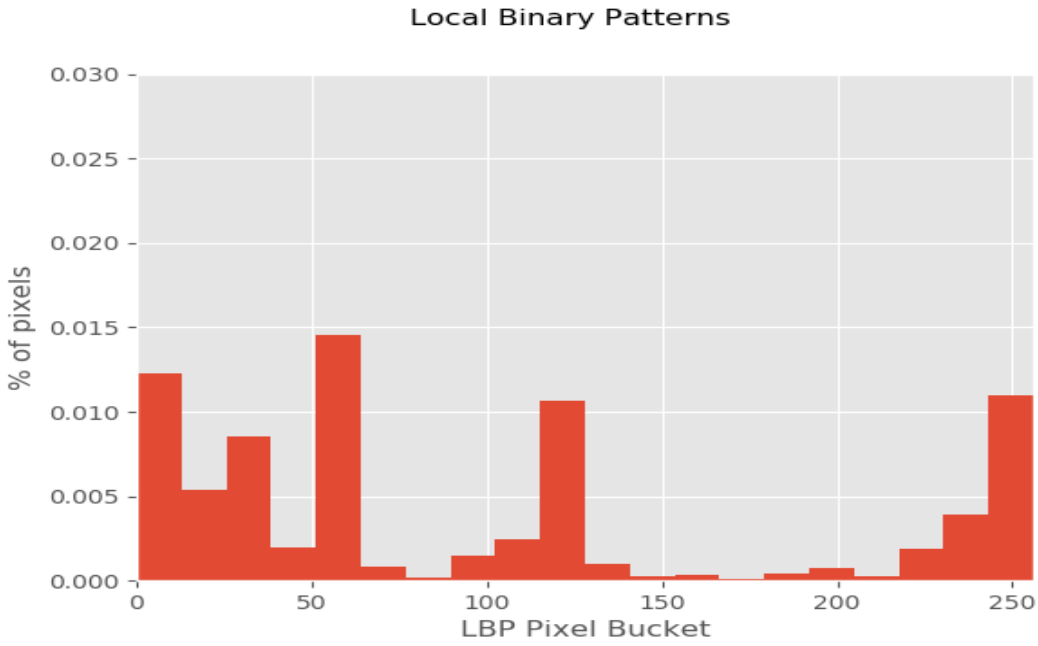


Figure 6-3: LBP Histogram

These figures are the LBP generated image (Figure: 6-2) and LBP Histogram (Figure: 6-3).

The LBP generated image is created by first converting the original image into greyscale image, as shown in the figure alongside of the LBP generated image (Figure: 6-2 Right), after undergoing LBP algorithm, as described earlier.

The Histogram is the quantified version of the face. In fact, it is pixel representation of a selected area. Here, the pixel percentage value is plotted against the pixel value, and the resulting plot is what is matched against the image scanned by the camera. The fact that the name was displayed by the screen when the face was detected, was due to the similarity of its histogram with one of the images stored for training.

6.3 Webpage Layout

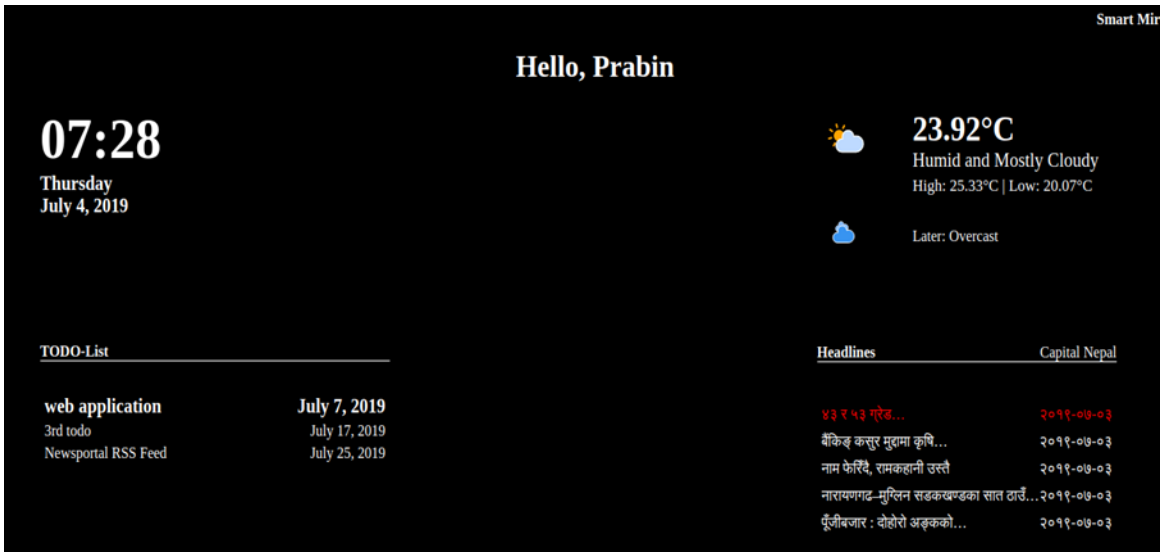


Figure 6-4: Webpage Layout

Above is the webpage layout. The webpage contains all the personalized information of the user, which are displayed systematically. Name of the user is displayed at the top. Time and date are displayed at the top left corner, below which is shown the To-do-lists. The central part is left blank for the user to have no obstruction while viewing face in the mirror. The top right corner contains the weather update and predicts weather condition for the ensuing seven days. Below the weather information is displayed the headlines from the preferred news portal, whose intensity fade as we move from the top headline to the bottom.

6.4 Error

There were not too many problems encountered during the course of the project. However, the problem of wrongly identifying a person for another, was, in a way, one too many for us.

6.5 Source of Error

The primary reason as to why we encountered the problem is the fact that not enough data had been trained.

As such, with enough training, the smart mirror has been performing satisfactorily compared to our expectations.

7 FUTURE ENHANCEMENTS

We might have achieved the proposed target, however, the project can be enhanced in terms for more personalized and exciting purposes. These can be listed as:

- A system can be built so as to support more visual/voice/gesture options.
- Using Kinect, we can add gesture recognizing applications.
- We can automate most of the steps to easily deploy/install the smart-mirror by simple end users.
- More add-ons to cover many use cases such as traffic, maps, booking taxi, etc.
- Styling apps can be added which can comment on the dress style or the hair style.
- Addition of health monitoring apps can be extremely beneficial.

8 CONCLUSION

As the project comes to the conclusion, we can safely say we have built a well-functioning **IoT based Smart Mirror with Face Recognition**.

Initially, we had hoped of a system that would recognize the face of the user, and would display the personalized information of the user on a webpage, which is what we have obtained at the completion of our project.

However, the project has not always been as easy as it may seem from the outset. Overcoming the error was a really challenging part. Also, the fact that we had to ingrain completely new concepts of web development, machine learning and image processing made this project a little too hard to get attuned to in the beginning.

As for the time, we managed to complete the project, well within the time frame set by the campus. However, the overall cost of the project just about exceeded the allocated funding provided by the University. However, it has to be noted that the two way mirror we had proposed earlier had been replaced, by a transparent and reflecting sticker which made the project a lot cheaper than what we had proposed earlier.

Apart from the mirror, there was another change in our project compared to the proposal. We had earlier expressed our desire to include speech recognition as well in the project, to which the teaching faculty advised us not to indulge into as it would be far more complex a project to deal with, given the time. Hence, speech recognition was excluded from the project altogether.

Overall, the project was more a learning curve than a daunting task, as it gave us an opportunity to get first hand acquaintance of concepts such as machine learning and web development.

9 APPENDICES

Appendix A: Raspberry Pi Specifications [9]

- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Dimensions: 82mm x 56mm x 19.5mm, 50g

Appendix B: Camera Module Specifications [8]

Still resolution	8 Megapixels
Video modes	1080p30, 720p60 and 640 × 480p60/90
C programming API	OpenMAX IL and others available
Sensor	Sony IMX219
Sensor resolution	3280 × 2464 pixels
Sensor image area	3.68 x 2.76 mm (4.6 mm diagonal)
Pixel size	1.12 µm x 1.12 µm
Optical size	1/4"
Linux integration	V4L2 driver available
Focal length	3.04 mm
Horizontal field of view	62.2 degrees
Vertical field of view	48.8 degrees
Focal Ratio	2.0

Appendix C: Camera Module Hardware Features [8]

Available	Implemented
Chief ray angle correction	Yes
Global and rolling shutter	Rolling shutter
Automatic exposure control (AEC)	No - done by ISP instead
Automatic white balance (AWB)	No - done by ISP instead
Automatic black level calibration (ABLC)	No - done by ISP instead
Automatic 50/60 Hz luminance detection	No - done by ISP instead
Frame rate up to 120 fps	Max 90fps. Limitations on frame size for the higher frame rates (VGA only for above 47fps)
AEC/AGC 16-zone size/position/weight control	No - done by ISP instead
Mirror and flip	Yes
Cropping	No - done by ISP instead (except 1080p mode)
Lens correction	No - done by ISP instead
Defective pixel cancelling	No - done by ISP instead
10-bit RAW RGB data	Yes - format conversions available via GPU
Support for LED and flash strobe mode	LED flash
Support for internal and external frame synchronization for frame exposure mode	No
Support for 2 × 2 binning for better SNR in low light conditions	Anything output res below 1296 x 976 will use the 2 x 2 binned mode
Support for horizontal and vertical sub-sampling	Yes, via binning and skipping
On-chip phase lock loop (PLL)	Yes
Standard serial SCCB interface	Yes
Digital video port (DVP) parallel output interface	No
MIPI interface (two lanes)	Yes
32 bytes of embedded one-time programmable (OTP) memory	No
Embedded 1.5V regulator for core power	Yes

Appendix D: Working of two way mirror

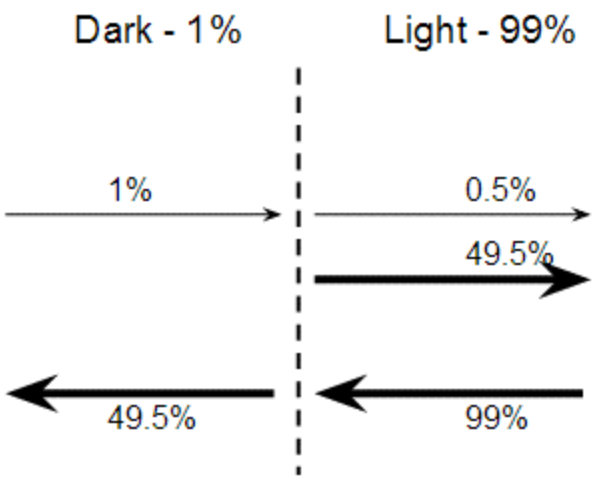


Figure 9-1: Appendix-Two way Mirror working

The working of a two way mirror relies on the difference between the amounts of light present in each side of the face of the mirror.

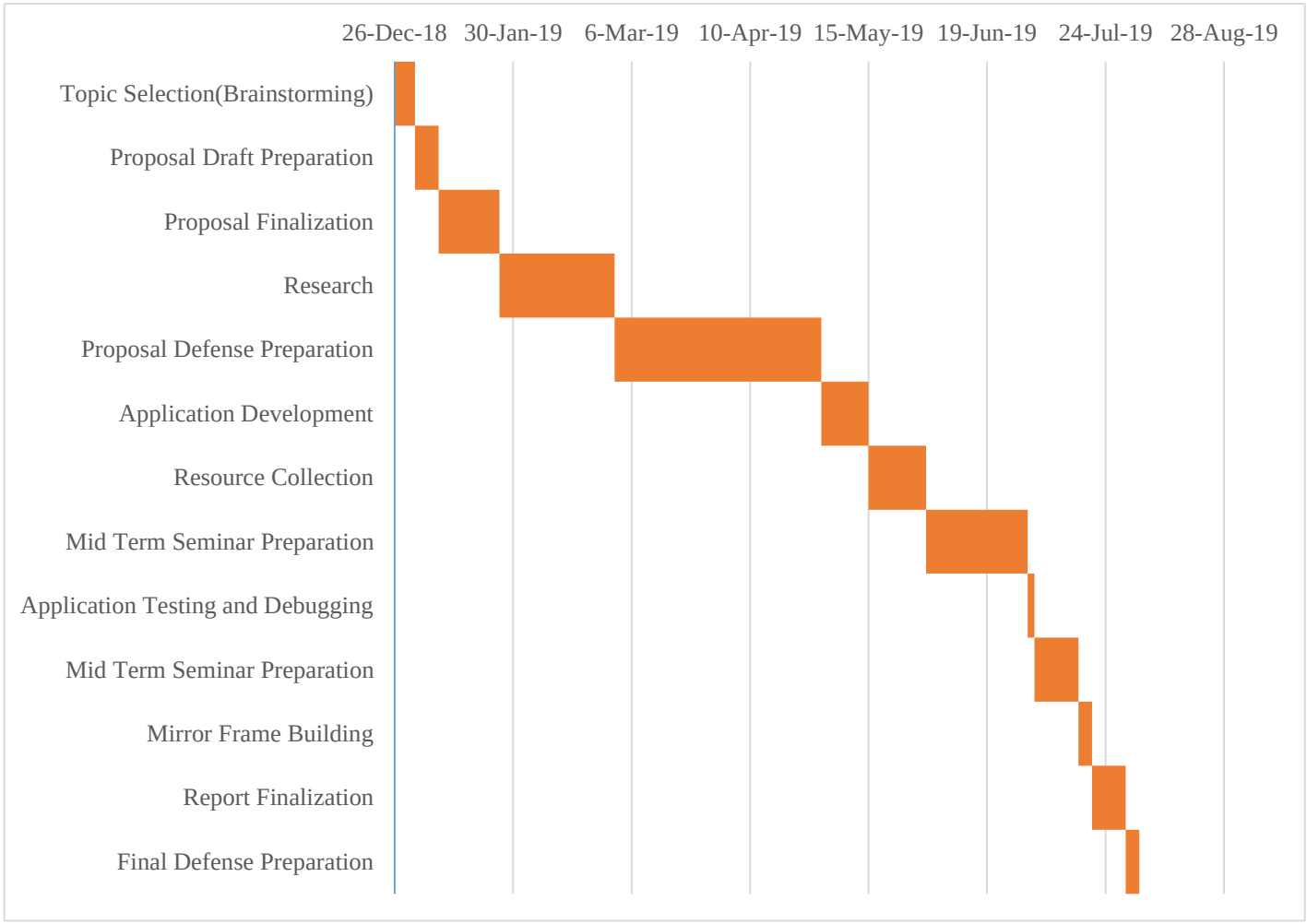
The above figure is a hypothetical scenario where the brighter part is 99 times as bright as the dark region. Assuming a 50-50 transmission-reflection case (although that is not the case in reality), the amount of light reflected back from the surface of the mirror to the brighter region is significantly high. Also, when observed from the dark region, the amount of light that transmits from the brighter region completely dominates that of the light reflected back to the dark region.

Hence, what we may conclude is that, an observer in the brighter region is not able to view beyond the surface, whereas one in the darker region is unable to view the reflection. [14]

Appendix E: Budget

Items	Cost
Raspberry Pi 3B+	6000
Camera Module	2500
Adaptor	1300
Micro SD Card	1000
VGA to HDMI Cable	700
Monitor	4000
Glass	1500
Frame	2500
Documentation	4000
Miscellaneous	3000
Total	26500

Appendix F: Gantt Chart



REFERENCES

[1] "Wikipedia," October 2011. [Online]. Available: https://en.wikipedia.org/wiki/Smart_device. [Accessed June 2019].

[2] "Internet of Things," 13 March 2019. [Online]. Available: https://internetofthingsagenda.techtarget.com/Internet-of-Things-IoT. [Accessed March 2019].

[3] "Wikipedia-IoT," May 2019. [Online]. Available: https://en.wikipedia.org/wiki/Internet_of_things. [Accessed June 2019].

[4] [Online]. Available: https://www.facefirst.com/solutions/mobile-face-recognition/. [Accessed July 2019].

[5] J. Q. J. Ruiz-del-Solar, ", “Illumination compensation and normalization in eigenspace-based face recognition: a comparative study of different pre-processing approaches,”," *Pattern Recognition Letters (PRL),* vol. 29 , no. 14, pp. 1966–1979,, 2008..

[6] "MagicMirrors," [Online]. Available: https://www.magicmirror.me/Home/History. [Accessed November 2018].

[7] "TAP Plastics," [Online]. Available: https://www.tapplastics.com/product/plastics/cut_to_size_plastic/two_way_mirrored_acrylic/558. [Accessed Jully 2019].

[8] "Raspberry," [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. [Accessed June 2019].

[9] "raspberry," [Online]. Available: https://raspberrypi.org/documentation/hardware/camera/. [Accessed July 2019].

[10] M. J. J. Paul Viola, "International Journal of Computer Vision," vol. 57, pp. 137-154, 2004.

[11] M. P. a. T. M. T. Ojala, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, p. 971–987, 2002.

[12] M. P. a. T. A. A. Hadid, "A discriminative feature space for detecting and recognizing faces," *Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR),* p. 797–804, 2004.

[13] C. S. M. A. Y. W. L. C. Di Huang, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey," pp. 1-3.

[14] "Mental Floss," [Online]. Available: http://mentalfloss.com/article/12969/how-do-two-way-mirrors-work. [Accessed Julu 2019].