An Integrated Vision-Based Architecture for Indoor Environment Security

Name of Candidate: **Fetlework Kedir Abdu**

A Thesis Submitted to the

Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the

Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia
September, 2019

An Integrated Vision-Based Architecture for Indoor Environment Security

Name of Candidate: **Fetlework Kedir Abdu**
Name of Advisor: **Prof. Yun Koo Chung**

A Thesis Submitted to the

Department of Computing

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the

Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

Adama, Ethiopia
September, 2019

DECLARATION

I hereby declare that this MSc Thesis is my original work and has not been presented for a degree in any other university, and all sources of material used for this thesis have been duly acknowledged.

Name: Fetlework Kedir Abdu

Signature: _____

This MSc Thesis has been submitted for examination with my approval as the thesis advisor.

Name: Prof. Yun Koo Chung

Signature: _____

Date of submission: September 23, 2019

APPROVAL OF BOARD OF EXAMINERS

We, the undersigned, members of the Board of Examiners of the final open defense by **Fetlework Kedir Abdu** have read and evaluated his/her thesis entitled **"An Integrated Vision-Based Architecture for Indoor Environment Security "** and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirement of the Degree of Masters in Computer Science and Engineering.

_____ _____ _____

Supervisor /Advisor Signature Date

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

ACKNOWLEDGMENT

Throughout the writing of this thesis, I have received a great deal of support and guidance from many people. First and for most, I would like to thank **Adama Science and Technology University** for funding me to undertake my MSc study. Then, I would like to express my sincere gratitude to my advisor **Prof. Yun Koo Chung** for his continuous support and guidance. It is with his follow up and encouragement this thesis came to a realization.

I also extend my acknowledgment and appreciation to **Dr. Mesfin Abebe**, who has willingly shared his precious time to provide supportive comments throughout the entire period of this thesis. I am indebted to his patience, motivation, enthusiasm, and immense knowledge that guided me well throughout my work.

I am also very grateful to my colleagues who had excellent cooperation helping me to collect valuable datasets. Last but not least, my special thanks also goes to my family for their continuous encouragement and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

2D	2 **D**imensional
3D	3 **D**imensional
ABS	**A**daptive **B**ackground **S**ubtraction
AFS	**A**djacent **F**rame **S**ubtraction
BGS	**B**ackground **S**ubtraction
CCTV	**C**losed-**C**ircuit **T**ele**v**ision
CNN	**C**onvolutional **N**eural **N**etwork
CPU	**C**entral **P**rocessing **U**nit
CVR	**C**omputer **V**ision and **R**obotic
DNN	**D**eep **N**eural **N**etwork
FBI	**F**ederal **B**ureau of **I**nvestigation
FPS	**F**rame **p**er **S**econd
GB	**G**iga**B**yte
GHz	**G**iga **H**ertz
GSM	**G**lobal **S**ystem for **M**obile communication
HMM	**H**idden **M**arkov **M**odel
HOG	**H**istogram of **O**riented **G**radient
IoT	**I**nternet **o**f **T**hings
IR	**I**nfra-**R**ed
KNN	**K**-**N**earest **N**eighbor
LBP	**L**ocal **B**inary **P**attern
LBPH	**L**ocal **B**inary **P**attern **H**istogram
MOG	**M**ixture **o**f **G**aussian

ORB	**O**riented Fast and **R**otated **B**rief
PCA	**P**rincipal **C**omponent **A**nalysis
PIR	**P**assive **I**nfra-**R**ed
RAM	**R**andom **A**ccess **M**emory
ResNet	**Res**idual **N**etwork
ROI	**R**egion **o**f **I**nterest
SIFT	**S**cale **I**nvariant **F**eature **T**ransform
SIG	**S**pecial **I**nterest **G**roup
SMS	**S**hort **M**essage **S**ervice
SSD	**S**ingle-**S**hot **D**etector
SURF	**S**peeded-**U**p **R**obust **F**eatures
SVM	**S**upport **V**ector **M**achine
USB	**U**niversal **S**erial **B**us
WSN	**W**ireless **S**ensor **N**etwork

ABSTRACT

A sense of personal and community safety is crucial to create a harmonized and productive society. However, the dramatic increase in home burglary activity is affecting social cohesion. Financial, physical and psychological impacts of such crimes are eroding trust within society and therefore lead to devastating consequences on social development. An automated security system for indoor environments is the best choice to address this severe problem. Broadly classified there are two types of automated security systems for smart homes: Sensor-based and Vision-based. Comparing with sensor-based systems vision-based home security systems are easy to set up, low cost and unobtrusive to catch illegal intruders in a wide range. Most of the vision-based systems developed so far are single-layered which can be easily deceived. This study proposes a novel integrated vision-based architecture to increase the security label (robustness) of a home security system. The system incorporates two modules: a face recognition module and a human motion detection module. A primary layer face recognition at the entrance door serves as a user authentication subsystem. A secondary layer human motion detection inside a room acts as a reliable backup in case, the primary layer is failed to catch the intruder for different reasons. The face detection and recognition is implemented using the current cutting-edge technology, deep learning. Histogram of Gradient (HOG) with Support Vector Machine (SVM) is employed to implement a human motion detection module. Several comparative experiments conducted have guaranteed a promising performance assuring the reliability of the methodological approaches followed to carry out this study. In addition, a scenario-based architecture evaluation with field tests conducted has shown the robustness and the feasible application of the proposed architecture in a consumer environment.

Key Words: Integrated architecture, Indoor environment security system, Face Recognition, Human motion detection, Vision-based system.

CHAPTER 1: INTRODUCTION

1.1 Background of the Study

We are living in a world where the burglary rate has been increasing dramatically. Indoor environments such as living houses and working places are the very first victims of theft initiated by unauthorized intrusions. Developing countries like Ethiopia are relatively insecure, many break-ins are reporting to a police every day while the security personnel is suffering from lack of evidence about a crime. This is because there is no reliable smart security system that can decrease the chance of getting broken into. Still today, people are using some manual tips to keep their homes safe. Arranging for someone who can look after a home, locking high-priced items and getting a barking dog around a house are the most commonly used traditional methods to deter robbery. These techniques provide a lower level of security compared with an automated in-house security system.

An automated indoor environment security system is very important because it is not only providing financial safety but also it creates a peace of mind for the users. An invader might cause a lot of crime including murder other than stealing. This severe problem that worth a human life makes an indoor security system become a hot research issue. For the last many decades, researchers have been developed various types of security systems to prevent unauthorized intrusion and to track illegal activities within the vicinity of highly restricted environments. Many of those previous systems are intensively sensor-based solutions [1]. Consequently, they are dependent on a substantial amount of contact with an input device and also their installation and maintenance cost is very expensive. As opposed to sensor-based systems, computer-vision based systems have many advantages to consumer applications. First and foremost, vision-based systems are unobtrusive. User authentication and intruder tracking can both be performed from a distance without any human intervention. Second, setup is easy and inexpensive as they only require simple low-cost vision devices of reasonable resolutions.

From computer-vision based home security systems that have been done so far, most of the researchers have focused on a single-layer system using face recognition on the camera mounted at the entrance door [1]–[4].

But what if the intruders entered through other means which is out of the camera field of view or a person wearing a mask comes and the face recognition system fails to prevent illegal intrusion?

This gap can be filled by developing a multi-layered security system architecture. Therefore, this paper proposed a dual-layered vision-based architecture for indoor environment security that integrates face recognition with a human motion detection module. Despite other researchers tried to integrate these two modules to secure an indoor environment, the way they did the integration cannot add a strong security layer. This is because they deployed the two modules to work dependently as a one-step authentication system; motion-triggered face recognition. If a person achieves to forge this system there is no secondary layer security.

Unlike the currently available architecture, this thesis did integration in a way that the two modules can work in parallel. The primary layer, face recognition at the entrance, acts as a user authentication system. And the secondary layer, human motion detection inside the room, acts as a reliable backup to detect and recognize human-related motion in case of failure in the primary layer. In both subsystems, efficient algorithms are used to make the security system more robust than before. Being integrated and vision-based the proposed system guaranteed the reliable result which can put its own print for the future label of security technology.

1.2 Motivation for the Study

The motivation for this research under this topic mainly arises due to the observation of a burning society problem related to criminal activities in indoor environments. Especially in the living houses, the crime is not only theft. Nowadays, there are a lot of news about a family murdered in their home, a girl raped in her living room and a child looted, etc. These news became a common scenario in our society. Having a reliable solution to this problem can prevent financial loss and many other crimes that are limiting social growth. The sense of security and peace that society gain using a reliable indoor security system is the greatest benefit of all. Next to being safe, confidence in feeling safe can make society to be more productive and harmonized. To provide this social benefit, researchers have been tried to develop security systems but there is still a gap to develop a reliable and robust solution.

Moreover, security is always a hot research issue that needs the researcher's contribution in different ways. This is because whenever the researchers find out a smart security system a person intended to violate security comes up with a new smarter method to forge a smart system. This fact grabs my attention to improve the current state-of-the-art under in-house security system using an Integrated Vision-Based Architecture.

1.3 Statement of the Problem

People can live in harmony and be more productive if only they feel secured wherever they are. But, the dramatic increase in break-ins rate in highly constrained areas is hindering this fact. Indoor environments such as living houses and working places have become the very first target of burglars that can lead to devastating consequences. The homeowners are losing their properties as well as getting uncomfortable (feeling insecure) whenever they are away from their homes. A lot of burglaries are reporting to a police every single day while there is no evidence for security personnel to use. According to FBI crime reporting statistics [5], there are roughly 2.5 million burglaries a year, 66% of those being home break-ins. Police solve only 13% of reported burglary cases due to a lack of evidence. The problem is even worse in a country, like Ethiopia, where there is no smart system for the homeowners to deter unauthorized intrusion. People are still using traditional techniques to keep their residential environment safe. Some of the most commonly used techniques are arranging a neighborhood who can look after a home, locking high-priced items, using metal bars on a fence and getting a barking dog around a house. These techniques can't effectively safeguard the property as automated systems do.

The home break-ins problem is severe than a material loss, it may even worth a human life. Even if, a burglar has no intention to kill a person there are incidents to happen, the owner might be at home during break-ins or come back when the burglar is inside. This situation might make the burglar emotional to take the worst action. [5] Stated that among the burglaries that occur during a person is home 26% of those people are murdered.

To solve all these problems, a lot of researchers have worked on automated security systems. However, the state-of-the-art of vision-based in-house security focuses on a single layer system, which is not reliable enough. These single-layered indoor security systems developed so far provide a one-time authentication at the entrance door. Most of these systems have used a face recognition technology to allow a person inside the restricted area [6],[8].

Once a person can bypass the face recognition system (might be by wearing a face mask or using another entry point out of the camera field of view) there is no other backup layer to be relied on.

Besides these vision-based systems, IoT based home surveillance systems have been developed since many years ago [10]–[12]. Such systems incorporate an enormous amount of sensor devices communicating with a network, might be unsafe, to perceive an environment. These sensors are with limited surveillance range which diminishes an unobtrusive nature of surveillance systems. Therefore, these systems are not only easily deceivable but also they are not affordable because of intensive hardware utilization.

All people value security and they need to install a system that enables them to keep their eyes on their properties wherever they are, even remotely. But the vulnerability and the high cost of the existing security systems are being the factor that limits the usability as well as the growth of this technology. This research tends to solve the above problem by designing a robust, reliable and relatively affordable architecture for indoor environment security systems.

1.4 Research Questions

The aforementioned challenges and problems related to indoor environment security should be solved by providing affordable and reliable solutions. And the gap in the research context needs to be bridged by developing a robust system. Therefore, this research is intended to get answers for the following central research questions:

RQ1: How much reliable and robust are the vision-based indoor security systems developed so far?

RQ2: What kind of integrated system architecture can be designed to make an in-house security system more robust?

RQ3: What kind of methods should be employed to implement the designed architecture so that the security system would be undeceivable?

1.5 Objective of the Study

1.5.1 General Objective

The main objective of this research is to design and implement a robust integrated vision-based architecture for indoor environment security.

1.5.2 Specific Objective

In order to achieve the main objective stated above, the following focusses will be specific objectives:-

- To design a robust vision-based architecture for indoor environment security.
- To prepare a dataset of face images for the face recognition module.
- To study and implement a current cutting edge technology, deep learning for the face recognition module.
- To develop a human motion detection subsystem with efficient algorithms.
- To evaluate the proposed architecture.

1.6 Research Methodology

In order to achieve the main objectives of the study and to answer the research questions, the following research methods are used. This section introduces the methods used in each module at the glance. The detail of the research methodology is explained in Chapter 3.

1.6.1 Face Recognition Module

This thesis adopted a pre-trained Convolutional Neural Network (CNN) model for face detection, and feature extraction stages. Whereas, for the label determination (classification) stage a machine learning classifier called Logistic Regression is used.

1.6.2 Human Motion Detection Module

Most of the previously done researches under motion detection used different sensors or at least 3D depth cameras like Microsoft Kinect. But this research is fully vision-based which doesn't require any hardware device except 2D camera; this is to exploit the advantage of

computer vision than sensor devices mentioned in subsection *1.1* of the document. Therefore, a highly accurate computer vision methods Adaptive Background Subtraction (ABS) is used for motion segmentation. And from the segmented motion region feature extraction is done using Histogram of Gradient (HOG) method. Then after, the extracted feature vector is fed to Support Vector Machine (SVM) to classify the segmented figure as Human or Non-Human.

1.6.3 Software and Hardware Tools

Different software and hardware tools are used to design and implement the proposed architecture. This work is implemented in python language with machine learning libraries like OpenCV, Scikit-learn and different frameworks such as Tensorflow and Keras. Jupyter Notebook and Visual Studio Code are code editors used for the implementation. For the documentation of this dissertation software like Microsoft Word and Microsoft Visio are used. A Samsung digital camera and a Logitech USB camera are employed for the dataset collection and model testing process.

1.6.4 System Evaluation Methods

To evaluate the complete performance of the designed architecture two evaluation methods are used. The first one is the model accuracy evaluation metrics and the second, software architecture evaluation method. To determine whether or not the designed architecture filled the gap of the current systems, a scenario-based evaluation approach is employed.

1.7 Scope and Limitation of the Study

1.7.1 Scope

The main focus of this study is to increase the reliability of the indoor security system by means of adding a security layer. Due to time constraints, the system is developed to detect a single intruder at a time excluding multiple intruders' detection for future work. And the research result is limited for only highly restricted indoor environments that are accessible by only authorized persons (like living houses, offices, computer laboratories, etc.); when no authorized person is available there. It is not intended for places where anybody is allowed to enter like shopping malls and banks. From highly restricted places, living houses are the most common victim for intrusion during day time. This is because living houses are left unintended for most of the day time. Statistical evidence [5] shows that if we take the burglaries of a day as 100%, then more

than 50% of them are committed during day time while people are at work. Therefore, this research is highly concentrated on day time vision excluding night vision security for future work.

Additionally, the notification subsystem is not in the scope of this research. Almost all of the previously done researches have incorporated immediate notification (such as buzzing alarm, streaming video and sending short message) for the owners or the security personnel at a time when any break-in is detected. An important point is to develop a robust architecture that doesn't let any unauthorized person inside the restricted area. Once we got such a system it is possibly easy to incorporate it with an existing notification system.

1.7.2 Limitation

The main limitation of this research is low-quality training datasets because of lack of high-resolution camera and well-organized environment for data collection. Moreover, for the face recognition module, the number of classes should be equal to the number of persons who have access to the indoor environment under consideration. This means that in the case of places where a lot of persons are allowed to enter like students' laboratories and large families' living houses the class size is very big.

The place taken for the experimental analysis of this research was a computer laboratory accessible by a few students, which means a small training data size. By its nature, machine learning is acutely sensitive to the quality and the quantity of data used to train the machine especially when the class size is big. Therefore, having small amount of the training set perhaps adversely affects the result of this study when it is applied to large class size problems.

Another factor that highly affected the success of this research is the problem of high-performance devices capable of deep learning. Deep learning is a computational intensive task that should be done using high-performance hardware like Graphics Processing Unit (GPU). This research is implemented using Central Processing Unit (CPU) which has a great impact on the response time of the system.

1.8 Significance of the Study

The result of the research is primarily beneficial to Adama Science and Technology University to keep secure its high-costed resources located in different offices and computer laboratories. Any indoor environments that are accessible for only legitimate users can also be benefited from the research finding. Additionally, the result of this research adds some knowledge of novel integration logic and methodological approach to the state-of-the-art of the subject matter. This study is a significantly great step to do further complicated security task including unusual activity analysis. In general, having the result of this study introduces robust technology to our country and employs it to solve extremely growing burglary problems. This is of great significance to promote social harmony and stability by providing smart technology for a better and smarter life.

1.9 Organization of the Thesis

This thesis is organized into seven chapters. *Chapter 1* introduces the concept and the focus of this work. This chapter provides the whole information stated throughout the paper in a precise and concise manner. *Chapter 2* presents a detailed literature review and related work. *Chapter 3* presents the methods, tools, and procedures used to accomplish this work. In this chapter, the rationale behind choosing the proposed methodology is discussed in detail. *Chapter 4* elaborates on the proposed novel integrated vision-based architecture for in-house security system. *Chapter 5* presents the implementation of the proposed system. *Chapter 6* discusses the experimental result of the proposed system. The last chapter, *Chapter 7* draws a conclusion based on the experimental result and recommends some future works on a vision-based indoor environment security system.

CHAPTER 2: LITERATURE REVIEW AND RELATED WORK

2.1 Literature Review

To assess what others have done in the area and to understand the problem better extensive literature review has been done on indoor security systems. Starting from preliminary review to the end of this work relevant literature such as books, journal articles, conference papers, manuals of development tools and resources from an internet that are related to a home security system were gone through. The review covers the background literature about the study topic, the recent progress on indoor environment security systems and the theoretical comparison analysis of different studies. Moreover, the review reveals the gaps that exist in previous works of literature and the significance of the proposed systems to bridge them. Works reviewed in this paper are organized chronologically and thematically as follows.

2.2 Overview of Indoor Environment Security System

An indoor environment is a residential and working space, as well as a place to store wealth. Break-ins into such environments have a lot of consequences ranging from financial to human life loss. Therefore, security systems become one of the mandatory consideration in keeping such environments from unauthorized access.

The modern solution for in-house security is a Closed-Circuit Television (CCTV) camera. CCTV is a device to monitor the real-time situation around the area it is installed such as office area, house, and building. It records each and every activity, and save the footage somewhere (it might be in the cloud, it might be in the device itself, depending on which one you buy). Sometimes, they also connect with your smartphone to stream live video what is going on at home, wherever you are in the internet-connected world [13].

One of the major problems with CCTV implementation is that it does not produce any notification and warning when it captures any suspicious object/activity. Because of this, CCTV provides a security solution with a time-consuming reviewing process even after the security violation has actually occurred. And also, CCTV captures the events that occur in the home environment continuously which results in huge consumption of bandwidth and storage media.

To counterfeit, these challenges researches have been done by integrating some features into a CCTV system. The paper [10], integrates motion detection and short message alert system into surveillance cameras to provide effective surveillance activity. Such trends surveilled indoor environments better than before. So, researchers have been rigorously participating to develop a system that can improve surveillance activities. Coarsely classified, there are two types of indoor environment security systems on which researchers have been actively contributed.

2.2.1 IoT Based Indoor Environment Security System

Internet of Things (IoT) is a network of interconnected electronic devices that are capable of sending data without interference or with the minimal human intervention [13]. This technology is widely used for smart city applications, transportation, manufacturing, health monitoring, and home automation. Some researchers have developed indoor security monitoring systems based on IoT concepts. Most of these researches intensively utilize the capability of electronics devices, e.g. Passive Infra-Red (PIR) motion sensor, glass break detector, door open sensor to monitor the occurrence of any anomalous activity. According to [14], the main components of an IoT based in-house intruder detection system are the following:-

- Sensors detecting intrusion through sensing physical environmental events like voice, temperature, vibration and objects' motion.
- A bidirectional data transmission media that is capable to transmit the detected signal to a local and/or remote operational control and command center as well as for transmitting commands of operators to the field devices.
- Processors for automatic evaluation of the data received from sensors.
- Workstations with some user interface software for operators to monitor alarms actuated by the sensors.

2.2.1.1 Common Sensor Technologies for Indoor Security System

IoT based indoor security system is an electronic security system that employs a lot of cooperating sensor devices. To cover the state-of-the-art technologies at the other side of vision-based in-house security it is important to see technologies familiar in IoT based home security system (see Figure 2.1).

A. Passive Infrared (PIR) Motion Detector

All living bodies and surrounding objects emit infrared electromagnetic radiation [11]. PIR sensor detects this heat radiation from objects which are moving in the detection range. Recently *Khirod et al.* [11], proposed IoT based intrusion detection system using a PIR motion sensor. This paper employed ZigBee to create a Wireless Sensor Network (WSN) and ESP8266 module to send data to a server located remotely. When a motion is detected sensor nodes that are implanted in every room send data to the center node. Different sensor nodes that use ZigBee for wireless transmission are all connected to a central node. In the notification subsystem, Global System for Mobile communication module (GSM) is used to send text alerts to the concerned user when an intrusion is detected.

Tanwar et al. [15], proposed an advanced IoT based security alert system in order to detect an intruder or any unusual event at home when nobody is available there. This home security system utilizes a PIR module and a Raspberry Pi for minimizing the delay during the process of e-mail alert. The PIR sensor generates a signal which is taken through the General Purpose Input Output (GPIO) pin attached with the Raspberry Pi. If the previous signal and current signal are same then there is no intrusion. Otherwise, there is a presence of intruders inside the home. In the second case, the camera attached to the Raspberry pi starts capturing the image and store it in the temporary storage available to the system. Finally, the system-defined mail can be sent to the homeowner.

Other researchers improved this system by adding a sample-based background subtraction algorithm called ViBe (Visual Background Extractor) to detect motion in red alert zone i.e. a place in the room where the valuables are placed [16]. One of the good things about the PIR sensor is an easy installation. But PIR based security systems create a lot of annoying false alarms. Sudden temperature changes like air turbulence can create moving objects that deceive a system to perceive as the occurrence of intrusion. Moreover, objects inside the detection area creating shadows can easily prevent motion detection leads to false-negative of intrusion.

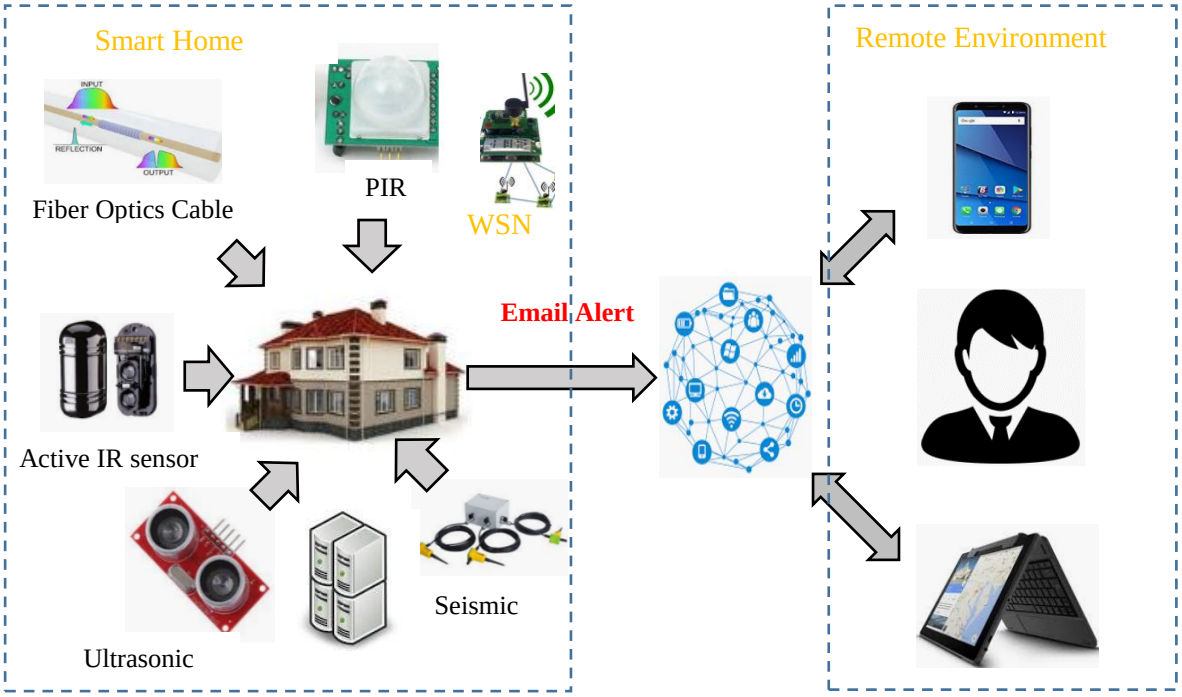


Figure 2. 1 Typical Architecture of IoT-Based Indoor Security System

B. Seismic Sensors

Some researchers developed an indoor security system by detecting seismic energy vibration created in the ground by walking or running above the detection area. A Seismic sensor has the ability to convert such energy into electrical signals. In-ground seismic sensors is a hidden installation which is suitable for surveillance. It also provides additional detection in case the vibration sensors mounted on the fence are bypassed by careful climbing.

The performance of the existing seismic sensor systems, however, is often limited by local interference sources [14]. To overcome this limitation, [12] developed Adaptive Noise Cancellation (ANC) algorithms that have proved effective results against certain classes of interference sources. In this work, the authors performed controlled field tests with interference from a nearby diesel tractor engine and gasoline power generator. The system performed well even under those conditions. The current research gap of this technology is it can be easily forged by a low-frequency signal.

C. Infrared Light Barriers (Active IR Sensors)

An active IR sensor is made up of two basic nodes, a transmitter, and a receiver. At one end of the protection zone, a transmitter node would be located generating a multiple frequency light beam to a receiver at the opposite end. A person or object interfering between the infrared of two nodes will be instantly detected. *P.Vigneswari et al*. [17], used an Active IR sensor to develop an automated security system for smart homes. In this system, the user gets an immediate alert when someone enters the room. The IR sensor employed to detect the persons entering the room and sends the output signal to a Raspberry pi board for processing. It also consists of GSM modem to send short messaging service so that the homeowner will be notified about the intrusion.

In [18], a motion detection system is developed using an active infrared sensor. According to this paper, Active IR sensor has the advantage of fast speed response of a relatively large sensor. This advantage permits simpler optical system design, especially for wide fields of view. On the top of large surveillance coverage, it is not sensitive to mechanical and acoustic noise, which presents substantial problems in the passive infrared (PIR) sensors. Low production cost is another advantage of these active infrared detectors. However, this technology requires a very clear line-of-sight between two units to avoid false activation of a security alarm. Therefore, it is not providing effective results in detecting security violence.

D. Active Ultrasonic Sensors

This sensor utilizes the Doppler principle to detect motion just like radar and sonar. They emit sound energy into a protected area and react with the change in the energy pattern. In 2013, *Shinu Yohannan et al.* worked on a home security system by using this sensor [19]. The main concern of their work was to design and implement a security system with an ultrasonic sensor module and enhance the system's reliability.

The ultrasonic sensor contains a transmitter and a receiver that is placed in a rotating motor. The assumption was that an ultrasonic sensor is set in a rotating motor to cover a wide range. The Ultrasonic transmitter periodically emits ultrasonic signals into an open area while the rotating motor can expand the coverage area into 360 degrees. If the signal ever hits any physical objects, it will be reflected back and captured by the receiver part of the sensor. The Micro-Controller Unit (MCU) will constantly check for the receiver output of the ultrasonic transmitter.

If the receiver output is high, the MCU will perform distance analysis of the object from the sensor using the fact that ultrasonic waves travel in air at 340m/s. Once the distance is calculated, MCU checks whether the object is within the range threshold specified for initiating the alert. If the object is within the range threshold, the MCU initiates a sound alarm and also the global GSM modem to notify the concerned person. In general, this sensor technology needs careful installation because there should not be a significant obstruction that affects the reflection between the sensor and the intruder.

E. Fiber Optic Cable Sensor

Fiber optic cable sensors use light for transmission and detection. The cable sensor can be fastened on the fences or can be buried underground. They register disturbance at a barrier (e.g. fence) or in the ground. The fiber optic cable must be bent or disturbed in some way, to affect the waveguide of the light being transmitted and thereby signaling a disturbance [14].

Tatsuya Kumagai et al. [20], used a Fiber-Optic Vibration Sensor for a physical security system. Like many other sensor-based security systems these authors install the fiber cable on the fence of the restricted area to detect a climbing intruder. To improve the reliability of the alarm, they have investigated a method that evaluates the frequency of the sensor output signal. According to the researchers' experiment, the vibration caused by a person trying to scale the fence is distributed in a wide frequency area. In contrast, the vibration caused by natural phenomena is in a narrow frequency area that is the sympathetic vibration of the fence. The system worked effectively at a time when software used in the sensor control unit can analyze the output values in a frequency domain.

An indoor security system equipped by such a sensor is good at surveilling relatively larger area. But some environmental impacts may cause a false alarm. The coolest side of this fiber optics cable sensor is, it can be used for transmission of communication data (e.g. video or image data) that live feeds a concerned user what is going on at home.

2.2.1.2 Challenges in IoT Based Indoor Environment Security System

In contrast with the traditional CCTV, IoT based security monitoring system equipped with a notification mechanism to warn the house owner if there is any intrusion. Even if, IoT could provide a better solution than the traditional CCTV, it has its own challenges listed below.

Cost

There are many commercial-of-the-shelf products of a home security system developed by exploiting the concept of sensor devices. The main problem is, this is something only a few can afford. The cost of ownership is way more than the purchase price, which might include maintenance charges, monitoring subscription fee if it is a monitored system, price of parts, battery replacement cost, etc. [21].

Dependency on the Internet

Internet is a basic requirement for IoT based systems. Without the Internet, IoT devices can't be cooperating together to achieve their purpose. Especially, in a country like Ethiopia where a strong Internet connection is a rare case, it is almost impossible to take advantage of such technologies.

Dependency on Professionals

High maintenance cost is another main problem in IoT based security protection of a smart home. In case there is some problem, only the hardware professionals can help to handle the problem.

Limitation of Surveillance Range

Most of the intrusion sensors used in IoT based home security systems have the capability of sensing some action over a particular small range only. This intrusive nature of the sensors leads the system to be highly vulnerable.

Falsely Triggered Alarms

This is where a security system's promise of **'peace of mind'** gets broken. IoT based home security systems should be helped with computer vision techniques to recognize the object that makes some motion. Otherwise, they will trigger frequent false alarms. It is very annoying that your security system triggered an alarm at midnight and wakes your entire family up, only to find out that it was a false alarm caused by your beloved cat or an unknown reason [21].

Security of Security Systems

In IoT based home security systems an electronic devices are connected through Wireless Sensor Network (WSN). If the security of wireless systems is not properly configured and enforced,

they can be accessed by attackers. This consequently led to a particular threat with the inception of smart home security systems, since this equipment can be accessed and control via the internet. In another scenario, signal interference can also trigger false alarms as alarming systems can take this signal for intrusion [21].

2.2.2 Computer Vision-Based Indoor Environment Security System

Computer vision-based systems deal with how computers can gain the ability to perceive the information in a digital image or video. Such systems try to mimic the human visual operation by using a camera. Many researchers introduced various types of computer vision-based systems for a better alternative of tackling the challenges discussed under IoT-based security system. The next section presents these works broadly categorized into two groups depending on the security layer they provide.

2.2.2.1 Single Layer Architecture

An indoor environment security system with single-layer architecture provides one-time authentication to get access to a restricted area. The literature of commonly used technologies that have been used to develop such security systems is discussed below.

A. Face Recognition

Recently, [6] proposed a security alert system using face recognition. In this study, the Viola-Jones algorithm and Principal Component Analysis (PCA) are utilized for a face matching decision. PCA works by finding the eigenvector of the covariance matrix of the set of face images. These eigenvectors are a set of features used to represent the variation between face images [22]. Using Euclidean distance the face captured via camera will be compared with database images. When the face is recognized as an unauthorized person SMS will be sent to the homeowner. PCA can be used with different classifiers like Euclidean Distance, Normalized Correlation, and Neural Network. Based on this research, the result of a comparative analysis of PCA with different classifiers is depicted in Table 2.1 below. The gap of this work is, it can only perform well if a person's face is upright frontal to the camera which is not the case during break-ins. The problem can be solved by using a face detection algorithm that is less sensitive to pose variation.

Table2. 1 PCA with Different Classifiers [6]

	Principal Component Analysis (PCA)		
	Algorithm and Classifier	Recognition Rate	Error Rate
1	PCA and Euclidian Distance	86%	14%
2	PCA and Normalization Correlation	75%	25%
3	PCA and Neural Network	70%	30%

In 2018, *Nashwan Adnan Othman* proposed a real-time recognition system to protect a restricted area [23]. Passive Infrared (PIR) sensor is used to detect movement in a specific area. In the presence of motion, a face will be detected and recognized in the captured image. Finally, the image and notification will be sent to the occupant using a telegram application. The Viola-Jones face detector is embraced for the face detection step. This study used the Local Binary Pattern Histogram (LBPH) algorithm for feature extraction step with K-Nearest Neighbor (KNN) classifier. The author could achieve 93.8±1.7 percent accuracy in different lighting and pose test images. In KNN input face will be compared with each sample in training set and the sample with minimum distance from the query image will be selected. All pixels in a face denotes unique data. So, it is computationally expensive for large databases.

In the mid-2017, a face recognition system was designed for a smart home application [8]. This research produced a system that could identify a person by using template matching and PCA. The system can only perform well in a condition: **a)** distance between the person and the camera is less than 240 centimeter, **b)** person doesn't use accessories that cover part of the face, **c)** Person's cloth and/or background color is different from skin color. Even under these constraints, it can only achieve 80% accuracy which is very low for security application.

Considering the complexity of the real-world situation, there would be prediction errors causing by the camera angle, illumination distance, background complexity and etc. [24] designed a face recognition system at a minimum requirement of reducing the effects of those factors. This article compares haar-like feature face detection with LBP and concludes that LBP can detect a face at a wider angle than haar- like features. So it used LBP to detect face and Support Vector Machine combined with Particle Swarm Optimization (PSO) to recognize it.

The authors did a comparative analysis on a performance of technique used for recognition and the resulted is tabulated in Table 2.2.

Table2. 2 Performance of Different Techniques used for Face Recognition[24]

Technique Used	**Recognition Rate**
PCA features-SVM	89.44%
LBP features-SVM	94.33%
LBP-SVM-PSO	96.54%

In mid-2015, *Ashraf Abbas et al.* [25] designed a security system based on facial expression recognition. This paper explained improving surveillance systems by detecting a person's expression which may exhibit the intention of doing criminal activities. The face will be taken by the camera inside a room to recognize once feeling. And if it is classified as criminal intention the system alerts security personnel. Facial expression systems can play a major role in preventing crimes before happening. But their accuracy doesn't fulfill the requirement for the security system. A person who is criminal can be easily detected as an innocent person and vice versa.

The efficiency of a home security system became improved with the advent of deep learning-based face recognition methods. *Giuseppe Amato et al.* [26], developed a facial-based intrusion detection system with deep learning in embedded devices. The authors performed face recognition using KNN classifier on features extracted from a 50-layers Residual Network (ResNet-50) trained on the VGGFace2 dataset. In their experiment, they determined the optimal confidence threshold that allows distinguishing legitimate users from intruders.

In general, an in-house security system based on a single face recognition technology can be easily forged to let the intruder inside the room. This is because a face recognition system might fail due to significant occlusion and pose variation burglars make to hide themselves. Besides, there are possibilities for a single face recognition system to ends up with a false Negative error that lets the intruder get inside without any trouble.

B. Motion Detection

An intrusion detection system is a challenging problem because of the variable poses and appearance of intruders to disguise themselves. The risk can be minimized by deploying motion detection systems instead of biometrics methods such as face recognition. In a way achieving this, a vision-based home automation system [7], is proposed to protect home reliably. Histogram of Gradient (HOG) feature descriptor and SVM classifier are algorithms utilized for accurate intruder detection. The good point is that the combination of these two algorithms can smartly avoid annoying false alarm arising from non-human motion. HOG uses Background Subtraction (BGS) to detect the motion region and then compute HOG feature descriptors by overlapping local contrast normalization [9]. Using those descriptors, SVM classifies motion to determine human presence. This system achieved 87.2% accuracy in detecting and classifying the moving intruder. This low accuracy is because of the illumination effect and can be improved by using adaptive background modeling during motion segmentation.

The article [27], proposed a real-time security system using motion detection. The Camera is used to catch the live images of the restricted area. The captured images are stored for further work. If a motion is found in this video, the computer will start recording, buzz an alarm and send SMS to people listed in its database. To detect the motion the research utilized the Adjacent Frame Subtraction method. This technique is not appropriate when a frame rate is low where the change of the motion is larger in successive frames. Consequently, system accuracy will be affected during night time. Adjacent frame subtraction doesn't give the shape of the object in motion which leads to difficulty for further processing. Additionally, this system doesn't have a way to distinguish authorize persons from unauthorized ones. Therefore, if an authorized person entered then also the alarm will be run.

The comparative analysis of the two known motion detection methods used in the above two papers is summarized as follows:

Table2. 3 Comparative Analysis of Motion Detection Systems

Author	Title	Publication year	Methods used	Accuracy	Gap
Danish Chowdhry et al.	"Smart Home Automation System for Intrusion Detection"	2015	✓ Background Subtraction ✓ HOG,SVM	87.2%	Sensitive to: ✓ Illumination ✓ shadow of moving object ✓ Reflection of apparent motion
Ahire Upasana et al.	"Real-Time Security System using Human Motion Detection"	2015	Preprocessing ✓ (RGB2Gray) ✓ Temporal difference, Thresholding to get the motion point	-	Difficulty in: ✓ Identifying object's shape resulting in difficult for further recognition. ✓ Identify stopped objects in the scene, false alarms.

2.2.2.2 Dual-Layer Architecture

The systems that are discussed in the above category are not robust enough. In the case of failure of such kinds of single-layer security systems, there is no backup subsystem to rely on. So other researchers have been considered module integration to develop a multi-layer security system that would be more reliable.

The smart house security system [28], developed a dual-layer security system by combining face recognition and automatic decision sub-system. In this system, the face recognition part is followed by automated decision making related to illegal intrusion. Automated decision uses artificial intelligence to calculate the probability of house resident's presence with respect to the parameter such as day of the weak, time of the day, the presence of owner's car, calendar holiday and electric consumption. This will be determined by neural networks. One problem with this system is affordability since it is an intensively sensor-based system.

Md Hazrat Ali et al.[29], proposed an integrated dual-level vision-based home security system, which consists of two subsystems **a)** movement detection and **b)** hand verification system. The primary movement detection technique is used to detect any movement first. If the system detects a moving human then it verifies the authority of the person for the secured place under consideration. It will check the threshold value where if the threshold level exceeds and the verification flag is off, the alarm will be triggered. Otherwise, if the verification flag is on, it means the person is authorized and movement detection will be turned off for this person.

In an event of a failure in the primary system, the secondary hand geometry verification module can act as a reliable backup to detect authorized persons in a restricted area. The second layer of this system requires the active cooperation of the user to show their hand in front of the camera which is not the case during an invasion. Such systems are not suitable for burglary detection because burglars always try to disguise themselves instead of present their biometric behavior for authentication.

Some researchers tried to combine the IoT concept with digital image processing. In paper [30], sensors are placed in the frame of the door to detect motion and to activate a camera. The camera sends the captured image to a machine responsible to recognize a face located in the frame. If the face is not recognized as authorized then the system alerts the concerned body through short message service. The image processing algorithms are considered to process the spatial and time complexity of the image captured to cross-check with the predefined dataset.

2.3 Related Works

In this section, the studies that tried to address the same problem by using the same methodological approach as this study are discussed.

Recently, [31] proposed a security system using a combination of face recognition and motion detection modules. This study focuses on the design and implementation of a low cost, smart and compact real-time monitoring home security system utilizing Raspberry Pi and OpenCV. The system detects the motion of an object as one subsystem. If a motion is detected then only the face recognition subsystem will be triggered. The face recognition method the author employed is Viola-Jones along with PCA. For the intruder's face which is not matched with any other in the database, the intruder's image and notifications will be sent to the concerned owners

as well as security officials. This research integrates two modules but the way it does integration can't add an extra layer on the security label. It is just a motion-triggered face recognition system.

Sameerchand Pudaruth et al. [32], developed a unified vision based intrusion alert system by integrating motion detection with face recognition. The motion detection module is responsible to determine the level of activity while the face detection module differentiates between authorized people and intruders. The motion level is calculated by taking the difference between the current frame and the previous frame. An alarm is raised when the motion level exceeds a certain threshold value (15%).

The paper [32] designed a system to capture 30 fps. If fewer frames are captured per second, some part of the action in the event may be missed. This means that the system is less responsive during night time where the infrared camera captures fewer fps in a darkroom. This results from the downside of Adjacent Frame Subtraction (AFS) method in the motion detection module. AFS is an appropriate technique for high frame rate sequences, where small changes of motions are often traceable. But for a low frame rate where there is a very small change in consecutive frames, it creates an erroneous motion image. Even under controlled conditions, the system achieves a recognition accuracy of only 62%. Thus, the authors conceded that additional work is required to make the system more robust and reliable.

John See et al.[33], proposed an indoor environment security system that combined the face recognition and motion detection modules. The author employed a primary and secondary processor for the purpose of detecting an intruder in highly secured areas. The system utilizes a simple analog camera at the entrance which is connected to a central server. The central server is a point at which preprocessing, feature extraction and recognition tasks take place. The central server also receives a video from CCTV cameras mounted inside a room to process it for motion detection. The face recognition module uses a simple template matching method with Principal Component Analysis (PCA). The motion is detected by the Background Subtraction method. The gap of [33], is triggering the alarm system whenever human motion is detected. The system didn't consider a way to identify whether a person created the motion is authorized or not. So, there is no doubt that the system suffers from annoying false alarms that run even for authorized person intrusion.

S*achin Patidar et al.* [34], proposed a vision based security system that detects unauthorized intrusion by using a change detection approach. In this research, image is captured from a web-camera at random intervals and frame subtraction is done to produce a final image with a major blob. And then, the area of the major blob is derived by calculating the mean of pixel array elements. These calculated values are compared with the theoretical value of a human's cross-section area. If calculated values fall in the range of theoretical values, it can be identified as a human intrusion. These ranges of theoretical values (the average cross-sectional area of a human) are defined in anthropology theory. This system has been tested for different environments, such as an experimental closed monitored area, and achieves over 86% detection rate at 5-6 fps processing speed. The main problem in this study is the dependency of the theoretical value on different factors such as camera position, the person distance from the camera and the camera angle from the floor.

The comparative analysis for the related works discussed above is summarized as follows:

Table 2. 4 Summary of Related Works

Author	Title	Methods Used	Accuracy	Gap
Khushbu Mehta et al.	"Vision-Based – Real-Time Monitoring Security System for Smart Home", 2016	✓ Viola-Jones for face detection ✓ PCA for face recognition ✓ Background Subtraction for motion detection	-	✓ Unable to detect face with bad pose and illumination ✓ Many false-positive motion regions
Sameerchand Pudaruth et al.	"A Unified Intrusion Alert System using Motion Detection and Face Recognition", 2013	✓ Viola-Jones ✓ PCA ✓ Adjacent frame subtraction for motion detection	62% motion recognition accuracy	✓ Not responsive to low frame rate input ✓ Face detection is sensitive to pose variation, illumination
John See et al.	"An Integrated Vision-based Architecture for Home Security System"	✓ Template matching ✓ PCA ✓ BGS ✓ Fuzzy-rule based classification	92% face recognition accuracy 82.8% motion recognition	✓ High false alarm rate ✓ Face detection is sensitive to even small occlusion
Sachin Patidar et al.	"Real-Time Vision-Based Security System",2018	✓ PCA ✓ BGS ✓ Anthropology theory	86% motion detection accuracy	✓ The result of motion recognition is dependent on the distance/angle of the camera from the floor

2.4 The Gap in the Current Architecture

As discussed in section 2.3, there are some studies done on vision-based integrated architecture for indoor security by combining face recognition with motion detection module. However, the way they did integration didn't solve the problem of highly deceivability of the system. The architecture uses one camera at the entrance door which stream frame to a server where motion detection takes place. If the motion is detected then only the face recognition module will be triggered. In other words, those systems are literally motion-triggered face recognition systems. The following flow chart represents the workflow of the current integrated vision-based architectures for smart home security.

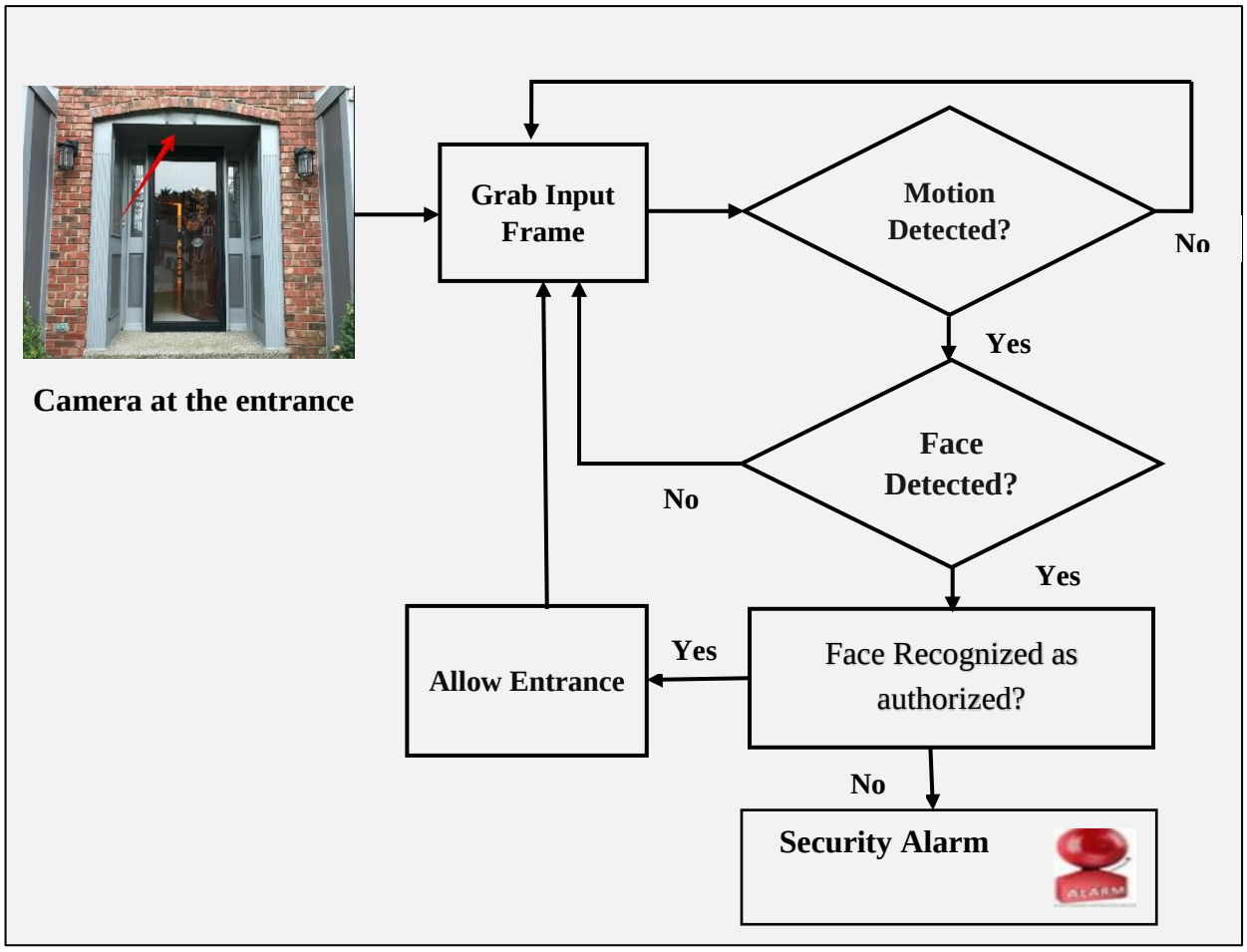


Figure 2. 2 System Work Flow of the Current Architecture

The question is, how much we can rely on the security system developed based on this architecture? One can definitely formulate different scenarios on which the system developed based on this architecture fails to secure the environment under consideration.

Scenario 1: An intruder with a face mask appears at the door.

In the case of this scenario, the system fails to detect the intruder as follows:

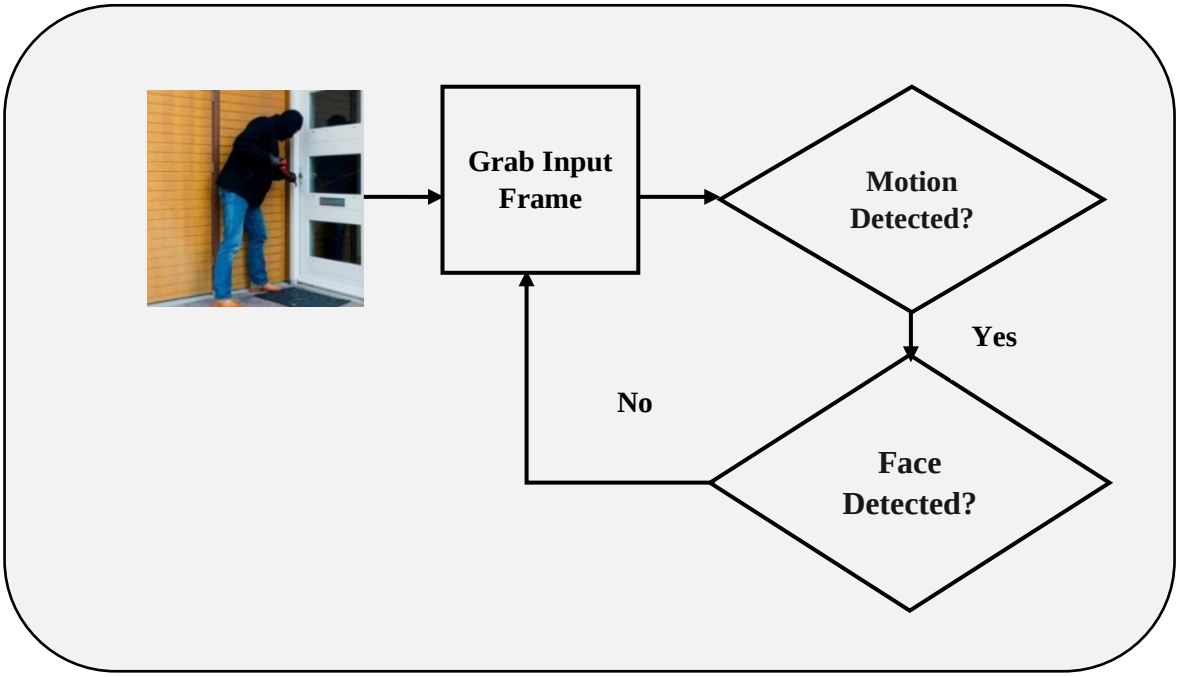


Figure 2. 3 Security Gap (1) of the Current Architecture

Scenario 2: An intruder already used another means out of the camera field of view to get inside the home.

In this scenario the system workflow looks like the following Figure:

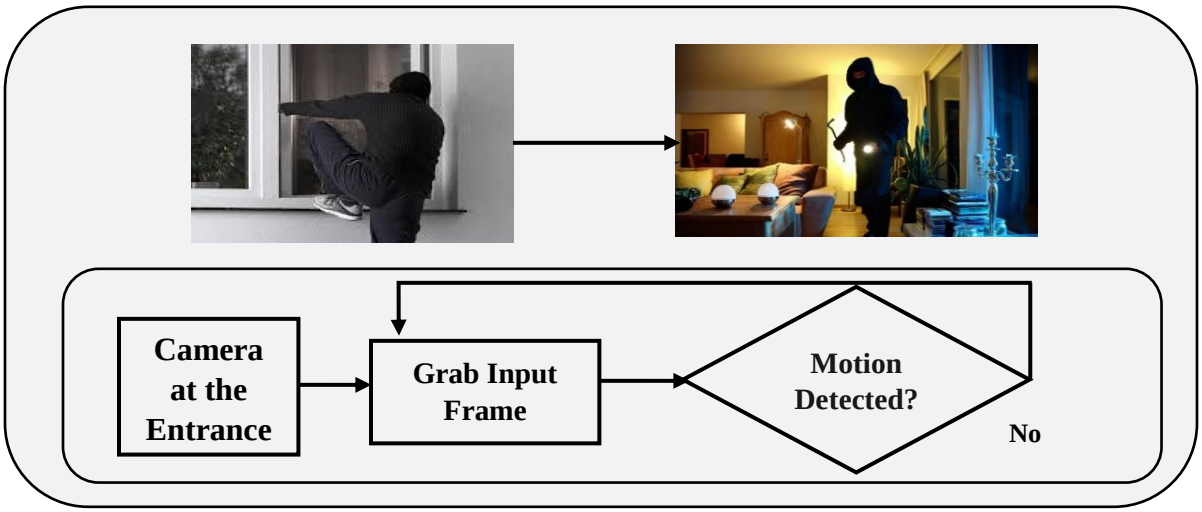


Figure 2. 4 Security Gap (2) of the Current Architecture

According to the above scenario-based evaluation of the currently available architecture, the system cannot reach out to the alarm system because it fails to detect an intrusion. The proposed novel system architecture of this thesis plays a vital role to address such problems.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Overview

This chapter elaborates on the detailed methods, tools, and procedures used for investigating the research problem. It provides a very concise background of a methodology that can fit the design, implementation, and evaluation of the proposed architecture. In more detail, it discussed about sample selection, data collection, and analysis method. Most importantly, this section contains the academic justification behind the method selection to achieve the main objective of the research.

The research methodology is highly determined by the nature of the research area and the purpose of the research in hand. Guided by the research questions mentioned in sub-section 1.4 of this document, the aim of this research is to integrate face recognition and human motion detection modules to develop a robust indoor security system. Therefore, this section elaborates:

- The data needed for the experimentation and from where those data are collected.
- The techniques used to improve the quality of the data.
- The best-fit methods employed to develop both modules.
- Suitable methods to evaluate the success of the research aim.

3.2 Data Collection and Preparation Methods

Despite a number of machine learning datasets available online, the dataset for this study is collected by the researcher. This is because the experimental analysis for this thesis took place in Computer Vision and Robotics (CVR) laboratory located in Adama Science and Technology University. This means that the machine should be learned by the dataset collected from the legitimate members to distinguish them from an unauthorized intruder. With this context, 75% of the training and testing dataset was collected by the researcher taking pictures and videos of the members who have legitimate access to the laboratory. As far as the data collection instrument is concerned, a Samsung digital camera is used to record videos for the data gathering.

3.2.1 Data Collection Participants

For the data collection, a purposeful sampling technique best fits this study. Within this context, participants were selected based on one pre-defined criterion: they should be a member of CVR SIG laboratory. This is because the idea behind this work is distinguishing those authorized members from unauthorized intruders. Accordingly, the participants for data collection were six students who are allowed to access the laboratory.

3.2.2 Data Collection Procedure

Data collection was held by recording 4-5 minute videos of the students mentioned above. For the face recognition dataset, the participants were asked to show different facial expressions and poses under different illumination conditions. This is to create diversified images that can effectively represent a particular person's face. The video contains a cluttered background and a human body other than the face region. Therefore, a region of interest (face part) had to be cropped. This is done programmatically by detecting faces in the video stream and save the cropped face images to a disk.

Again, another video of participants walking with different styles is recorded. From these videos, a dataset used for the human motion detection module is collected. Human motion detection is a binary classification to discriminate a motion belongs to human being from a motion of other objects. Therefore, to implement a human motion detection module a machine should be learned with both human (positive training examples) and non-human figures (negative training examples). Negative training examples are sample images that don't contain an object of interest which is gathered from one of the biggest computer vision online databases, ImageNet [35].

3.2.3 Data Preparation Methods

The dataset which was gathered from the field was not clean. Most of them were contained motion blur, shadows, and unacceptable occlusion. So, data cleaning was done manually to remove those irrelevant images from the dataset. After cleaning, a digital pre-processing (such as resizing, gamma correcting, aligning and smoothing) was applied to enhance the data for

further processing. After this, the dataset is categorized into their corresponding classes to be used during model training and testing.

3.3 Software and Hardware Tools

3.3.1 Programming Tools

Different software and hardware tools were employed to implement the proposed architecture. The system was developed in python language using a Visual Studio Code (VSCode) editor. An open-source web application called Jupyter Notebook was in a role for data preparation and visualization. The main reason to pick Python as an implementation language is that it's simplicity and access to grate machine learning libraries. Different libraries and frameworks discussed below are utilized in this study. Python with its technology stack made these packages reusing a lot cheaper and easier.

3.3.2 Machine Learning Libraries

Among many, OpenCV is one of an open-source computer vision and machine learning Library. It was built to provide a common infrastructure for computer vision applications. This library has more than 2,500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms [36]. To accelerate the use of machine perception in the subject matter, this study intensively used OpenCV and its pre-trained deep learning model.

Besides a pre-trained feature extractor, this study incorporates various classification stages. This is a point where scikit-learn came into play to use the classification method such as Support Vector Machine (SVM) and Logistic Regression. For scientific computation, this research employed the fundamental package with a multi-dimensional array object known as Numerical Python (Numpy). The famous deep learning python frameworks, Tensorflow and Keras, are also used to interface some pre-trained models.

3.3.3 Hardware Tools

Hardware tools such as Intel Core i5 CPU 2.40 GHz, 4.00 GB RAM machine, 2 Megapixel Samsung camera and 2 Megapixel Logitech USB camera were used to carry out the laboratory experiments.

3.4 Methods for Face Recognition Module

Through the years, many algorithms and techniques have been used to recognize faces, such as the Eigenface of Principal Component Analysis (PCA), fisher face method, frequency domain analysis, Hidden Markov Model (HMM), etc. However, the accuracy provided by those algorithms is not satisfactory. The one that is currently used, and providing the best results is Deep Learning. Because of the need for highly accurate results this research will focus on the currently available cutting edge technology, deep learning. From widely used deep learning architectures a Convolutional Neural Network (CNN) is used, which best fits the face recognition system.

Depending upon the amount of training data in hand CNN architecture can be used in various approach [37]:

1. Large data set, **model training from scratch.**
2. Less amount of data, still a good number of images per class **fine-tuning a network. (Transfer learning)**.
3. Very less amount of images, adopt any pre-trained network to convert the target images into a feature vector and then train some ML model on these feature vectors and use it for classification. **(CNN Code approach)**

In the case of this work, the amount of data for face recognition is very less (a small number of authorized users for the restricted environment). So, the study took the advantage of CNN code approach and adopted the pre-trained Deep Neural Network (DNN) model for face detection and feature extraction step. The feature vector extracted by the DNN model is fed to logistic regression for label determination. The overall methods used in this module can be divided into four main steps as most face recognition schemes: - face detection, preprocessing, feature extraction and classification (see Figure 3.1).

3.4.1 Face Detection

Deep learning-based face detection outperformed various traditional methods like Viola-Jones, Local Binary Pattern (LBP), Template Matching, Histogram of Gradient (HoG) and many others. Face detection techniques should provide accurate detection which is not sensitive to factors like pose variation, illumination change, occlusion, and cluttered background.

Especially, for indoor security systems pose variation and occlusion have been the most challenging factors. This is because burglar tries to disguise their faces with different bad possess.

To counterfeit, this challenges a face detection model trained with an enormous amount of data should be employed. So, this study implemented a highly improved deep neural network-based face detector trained on ImageNet dataset. This face detector is based on a Single Shot Detector (SSD) framework with ResNet-10 as a backbone base network [38].

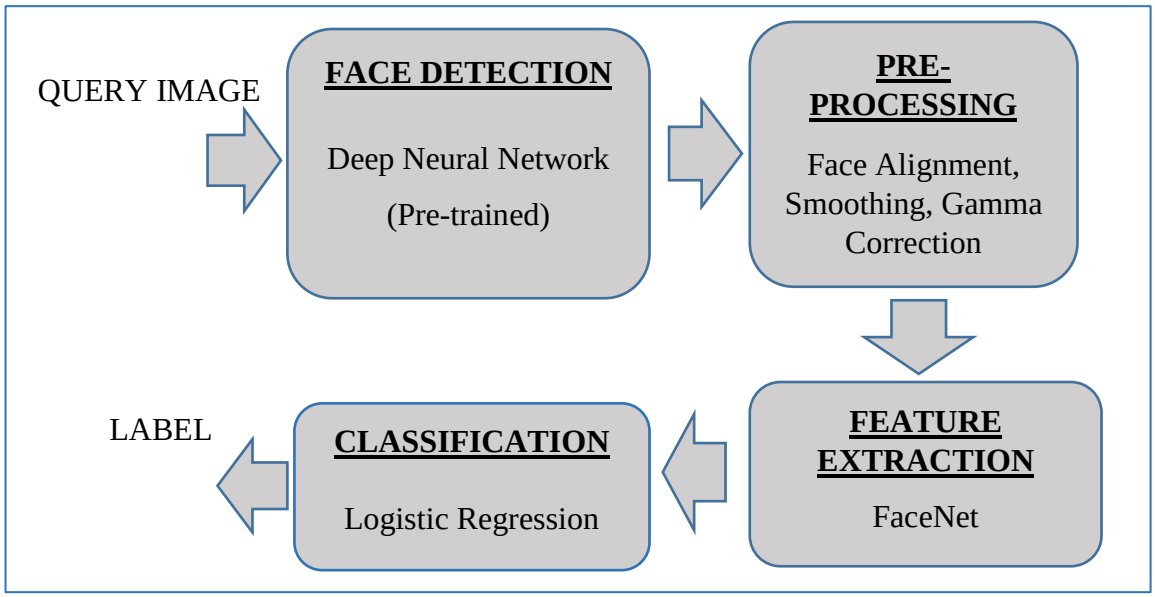


Figure 3. 1 Proposed Methodology for Face Recognition Module

To obtain correct prediction from a deep neural network the input data should go through a pre-processing step. To pre-process the captured image the DNN face detector by itself implemented mean subtraction, scaling and color channel swapping. There are a lot of approaches to normalize an image but mean subtraction is very powerful to combat illumination changes in the input images [39]. Mean subtraction computes average pixel value across all images in the training dataset for each of Red, Green and Blue channel. Subtracting these values from original pixels allows each feature in a similar range so that the machine learning process will be controlled.

3.4.2 Pre-processing

To increase the recognition capability of the face recognition model this study applied some sort of pre-processing to the data before going for recognition.

A. Face Alignment

Face alignment is a data normalization technique for identifying the geometric structure of human faces in digital images [40]. Applying this method is important to obtain a normalized translation, rotation, and scale of a face image. It is very common to align a face before training the recognizer to gain good accuracy. There are many forms of aligning technique but most of them are based on a complex 3D model transformation that increases a computational time. This study applied alignment by using a simplistic method rely only on facial landmarks.

B. Gamma Correction

Each pixel of an image has its brightness label (luminance). Imaging devices have a limitation of non-linearity to correctly capture luminance. Due to this fact, the image collected for this study had various distortion. Therefore, they have passed through a gamma correction process because face recognition results may be affected highly unless the image disturbance is removed.

C. Smoothing

The data collected from the field were full of image noise (random variation of brightness) which has a huge effect on recognition accuracy. To reduce this noise the Gaussian Smoothing technique was employed. The working principle of the gaussian blur is convolving an image with a gaussian function. Because of this, it is very fast compared with other state-of-the-art smoothing techniques (like optimization-based smoothing, gradient weighted filter, median filter and etc.) [41].

3.4.3 Feature Extraction

The traditional machine learning approach extract features from images using global feature descriptors such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HoG), Color Histograms, etc. In the case where global descriptors are not suitable, there are local descriptors such as Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF),

Oriented Fast and Rotated Brief (ORB), etc. These are hand-crafted features that require domain level expertise.

Instead of using hand-crafted features, Deep Neural Network automatically learns highly descriptive features from images in a hierarchical fashion. To take this advantage Face Net deep convolutional network, originally proposed by Florian Schroff et al. [42], is employed in this study. The FaceNet network is trained in such a way that the face image is mapped with embedding space that corresponds to face similarity. It is an appropriate method for fast training even on resource-limited devices. This is because it resolved the downside of previous deep network problems: very large representation size. In contrast with other previous networks, FaceNet can represent a face with only a 128-D compact vector, by using triplet loss illustrated in Figure 3.2.

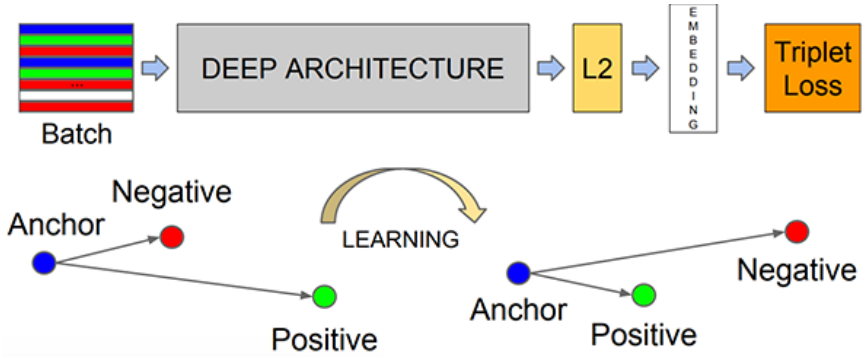


Figure 3. 2 Learning Principle of FaceNet (Triplet Loss)[42]

3.4.4 Classification

Deep learning models are layered architectures that learn different features at different layers. This layered architecture allows us to utilize a pre-trained network as a fixed feature extractor model with a machine learning technique as a final layer. This type of approach is well suited for this study because of the limited data size. Therefore, Logistic Regression is used at the classification stage of the face recognition module.

3.5 Methods for Human Motion Detection Module

Most of the existing human motion detection systems deployed a motion sensor to detect some movement. However, this thesis proposed a fully vision-based architecture for indoor

environment security. So, the motion detection module is implemented using efficient vision-based methods without any motion sensor, see Figure 3.3 below.

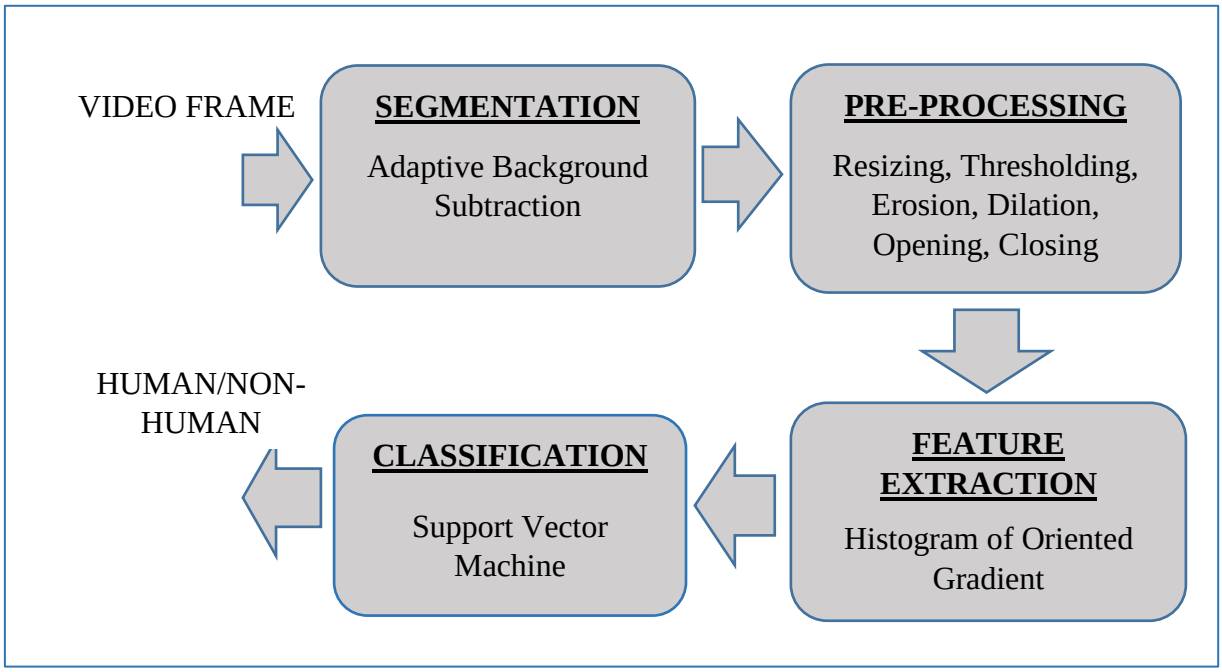


Figure 3. 3 Proposed Methodology for Human Motion Detection Module

3.5.1 Segmentation

There are various approaches for detecting motion in a video stream. Most of the approaches follow a similar strategy that is by comparing the consecutive frames or simply comparing the current frame against the background frame. Adjacent Frame Subtraction (AFS) and Simple Background Subtraction (BGS) are the most commonly used methods in motion segmentation. But both of them have their own downside.

- Adjacent Frame Subtraction simply produces the difference image, $D(x,y,t)$ between an input frame, $F(x,y,t)$ and the next successive frame, $F(x,y,t+c)$ after a fixed time interval c [43].

$$\mathbf{D(x,y,t) = |F(x,y,t) - F(x,y,t+c)|} \qquad \text{(Equation 3.1)}$$

This method can't extract the silhouette of a moving body especially in a low frame rate where the change of motion is large in a successive frame.

- Background Subtraction calculates the difference image, $D(x,y,t)$ between a background image, $B(x,y,t)$ and the currently captured frame, $F(x,y,t)$ [43].

$$\mathbf{D(x,y,t) = |F(x,y,t) - B(x,y,t)|} \qquad \text{(Equation 3.2)}$$

Simple BGS is not accurate when the object is moving fast where there are relatively small changes in motion due to the frame differencing. In addition, this technique collects the first few frames to generate a background image to be subtracted from each and every current frame. Therefore, this method is not robust against sudden background change. Consequently, it is difficult to get an accurate moving object using a simple BGS technique. Come to the proposed security system, the motion detection method should provide an accurate foreground mask since the motion part of the frame is used for further authentication. To achieve this and to compromise the aforementioned flaws, this study used the Adaptive Background Subtraction Method.

Adaptive Background Subtraction technique composed of two steps: background modeling and background updating [44]. Unlike simple BGS, this method initializes the probability background model by the first frame, updates it and extracts foreground objects at every frame. Therefore, this method automatically adapts to the environment to extract a moving object in a precise manner.

3.5.2 Pre-processing

After Segmentation, the foreground image contains false information due to noise in the video. In order to remove this noise and to extract a clear Region of Interest (ROI) morphological operations: erosion followed by dilation, opening, and closing is applied.

3.5.3 Feature Extraction

From the foreground motion image of a video frame features should be extracted for further recognition of an object caused the motion. For this step Histogram of Gradient (HOG) [9] is used for its high accuracy record in human detection. The basic reason to pick HOG instead of many other feature extractor is its ability to extract both shape and texture information of an image. This is a great advantage to recognize a human figure enclosed in motion region of a given frame.

3.5.4 Motion Classification

The classification step in the human motion detection module employed Support Vector Machine (SVM). SVM is a suitable machine learning algorithm for binary classification [45], as in this study it decides whether the foreground motion image contains a human figure or not.

3.6 Evaluation Method

After the implementation is conducted the experimental methodology and the result obtained are evaluated against the research objective. To evaluate the complete performance of the designed architecture two evaluation methods are used. **A)** Model accuracy evaluation metrics with the K-fold cross-validation method. Among the most common accuracy measurement technique, this study employed a confusion matrix. **B)** Software architecture evaluation method. To determine whether or not the designed architecture satisfies the requirement in hand (robustness), a scenario-based evaluation approach is employed. The detail of experimental implementation and system evaluation is presented in Chapter 5 and 6 respectively.

CHAPTER 4: PROPOSED SYSTEM ARCHITECTURE

4.1 Overview of the Proposed Architecture

The main objective of this study is to fill the security gap of the current indoor environment security system by designing a dual-layered architecture. To achieve this, the proposed architecture integrated two vision-based modules, **A)** face recognition module and **B)** human motion detection module. This architecture makes a security system more robust by adding a security layer. The primary layer, face recognition at the entrance authenticates the user. However, this module could fail to detect unauthorized intruder because of many reasons. In this case, the secondary layer, human motion detection inside the room serves as a reliable backup. Both modules work as a multi-step authentication system running on the same machine parallelly (see Figure 4.1)

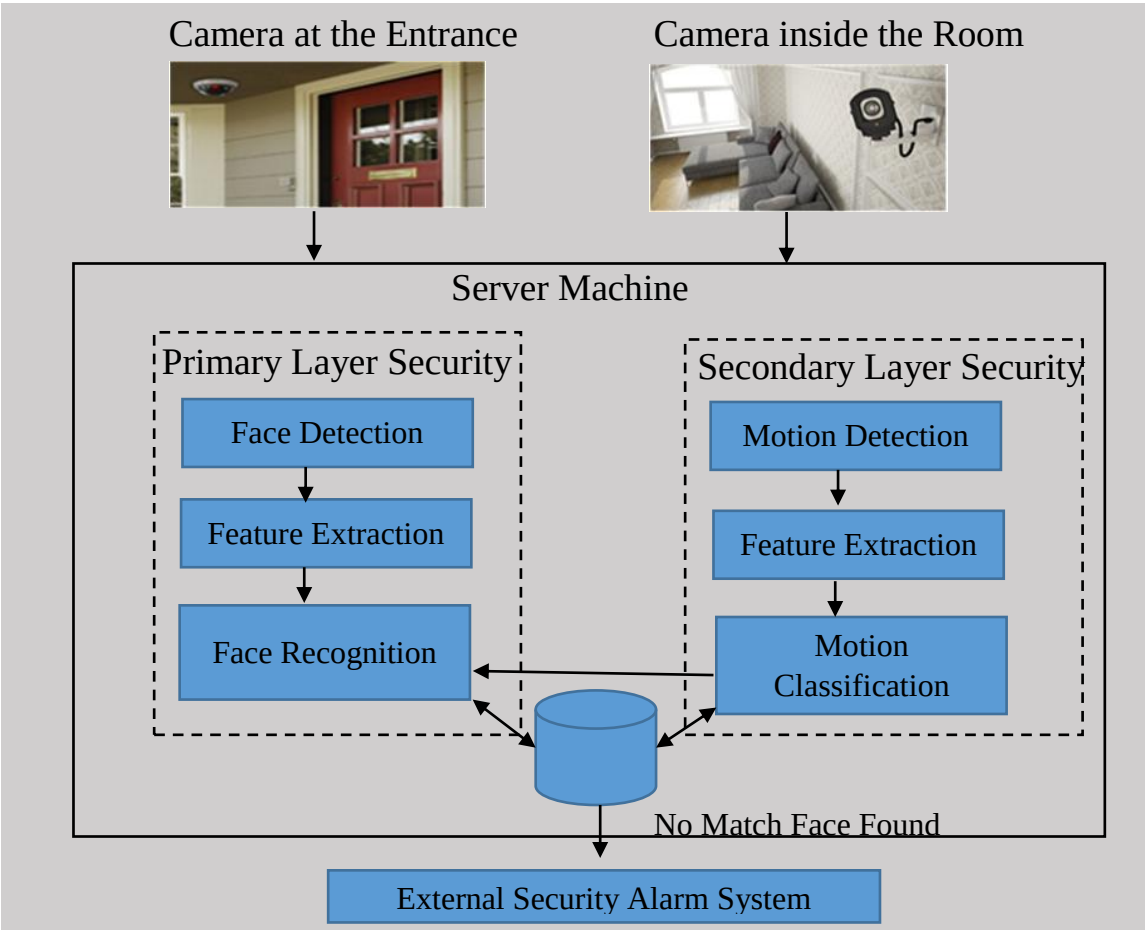


Figure 4. 1 Proposed System Architecture

The detailed workflow of the security system developed based on the above architecture is depicted in Figure 4.2, the system starts by capturing an image using a camera mounted at the entrance door. From the captured image face recognition will be done on a server machine. If a face is recognized as authorized it allows the entrance otherwise, it sends a signal to an external notification system. In case, an intruder comes wearing a mask and got the chance to enter the room because of face recognition failure, the proposing system provides an additional security layer inside the room.

The CCTV camera mounted in the room captures consequent frames. The frames will be analyzed to detect if a human intrusion has occurred. Following the motion detection, motion classification will be done to distinguish a human motion from other destructive objects/pet motion. If only the motion is classified as human motion then face detection will be triggered for further verification of a person who caused a motion. No face detection at this moment means that the person tried to disguise his face by using a face mask. So, the system should send a signal to initiate external alarm system that can notify the homeowners about the intrusion.

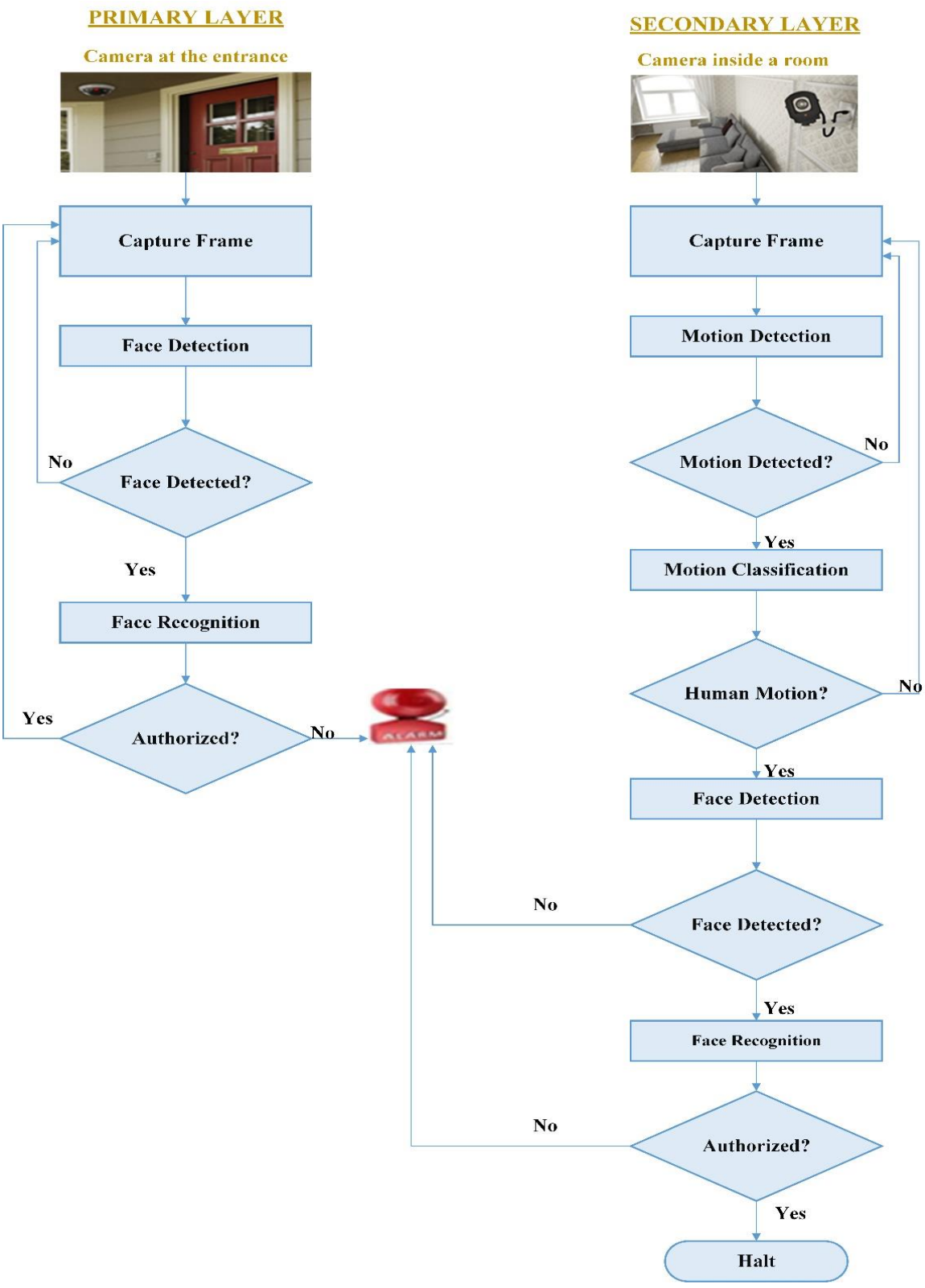


Figure 4. 2 Workflow of the Proposed System Architecture

4.2 Proposed Sub-systems Architecture

A. Face Recognition

This primary layer functions as the user authentication subsystem to grant entry into the home to authorized persons. This module plays a vital role to deter a burglary before it actually occurred. The state-of-the-art methods are applied for this module to make the recognition robust. Like most of a face recognition implementations, this module contains four main steps (Face Detection, Pre-processing, Feature Extraction, and Classification) as depicted by the following block diagram.

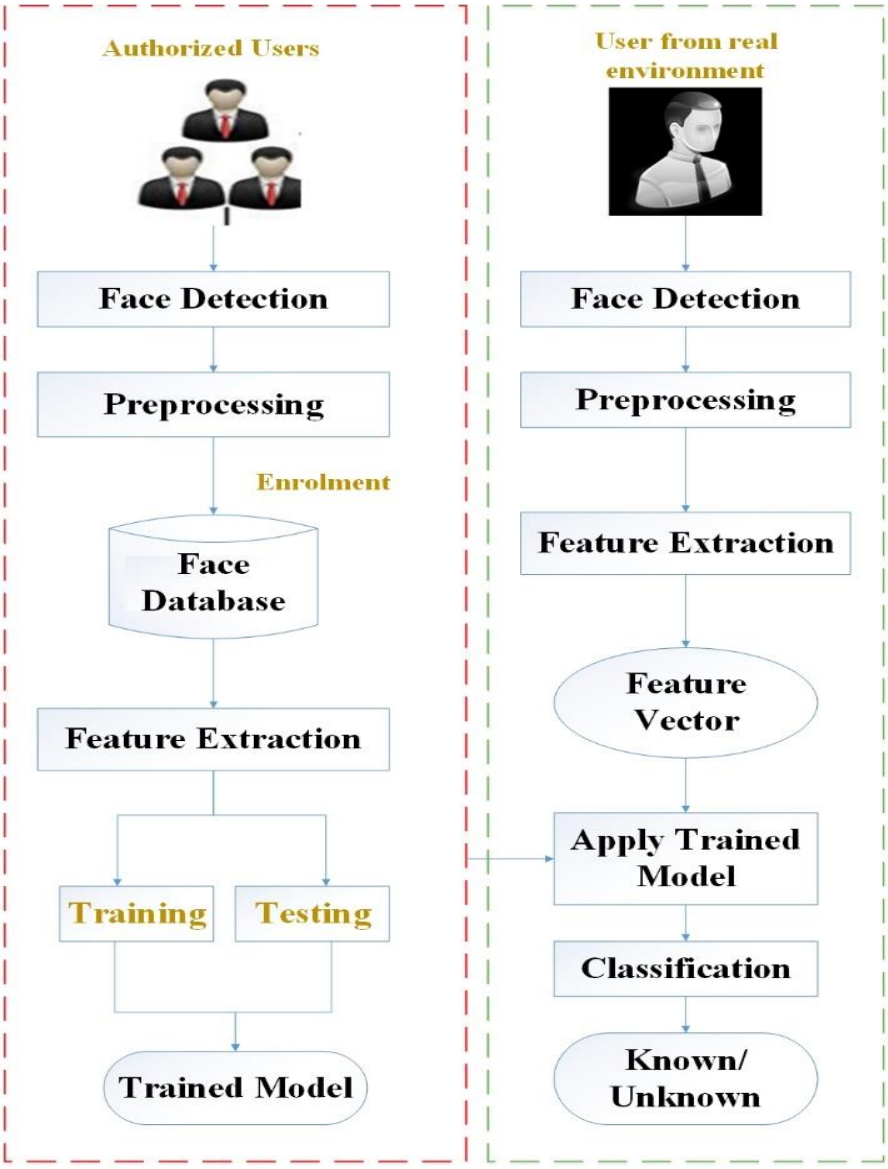


Figure 4. 3 Architecture of Face Recognition Module

B. Human Motion Detection

Human motion detection is a process of recognizing a human being regardless of its identity, based on certain characteristic patterns or features of the exhibited motion [46]. In many computer vision problems, the detection of human motion is being used as a first step for more high-level problems such as head tracking, gait recognition, and unusual activity recognition. In this work, human motion detection is incorporated to detect an intruder inside the room as a backup security module. This module aims at segmenting the motion area of a moving object and classifying it as a Human or Non-Human entity. Human or Non-human classification is essential to filter out annoying false alarms due to the motion of non-human entities such as pets. Figure 4.4 shows the architecture of the human motion detection module.

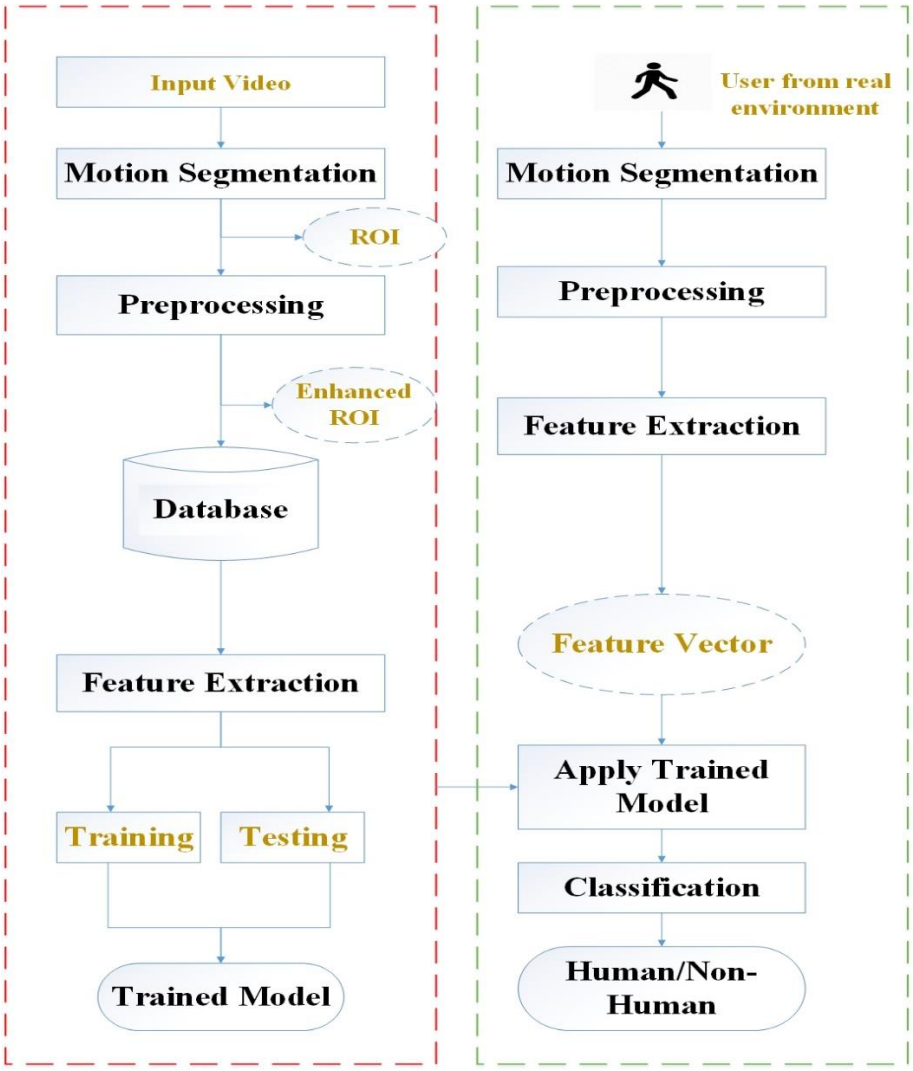


Figure 4. 4 Architecture of Human Motion Detection Module

CHAPTER 5: IMPLEMENTATION

5.1 Overview

This chapter presents the detailed implementation of the proposed system architecture. The system is developed in python programming language under the experimental setup shown in Table 5.1 below. As clearly stated in Chapter 4, the proposed study integrates two modules that can function in a parallel fashion. Therefore, the implementation process of both modules is discussed separately step by step in the following subsections. Both modules share the common computer vision implementation pipeline: Detection, pre-processing, feature extraction, model learning, and testing.

Table 5. 1 Experimental Setup of the Study

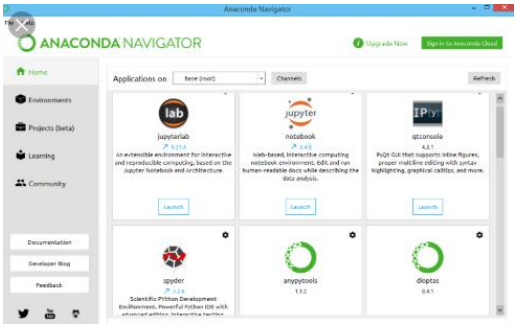
Experimentation Environment

Hardware Setup

1080x1920p camera for data collection

Logitech USB camera Connected with a Laptop for testing

Software Setup



Code Editors

Libraries and Frameworks

- Python 3.6.7 64-bit
- OpenCV 4.0
- Scikit-Learn
- Tensorflow 1.12
- Keras

5.2 Module Integration Logic

The high-level description of the way how the modules are integrated to function together in a parallel manner is given below.

Pseudocode

INPUT: Video frame from camera 1 (camera at the entrance) and camera 2 (camera inside)

OUTPUT: Identification of a person in front of a camera and a motion class

PROCEDURE:

BEGIN

Initialize video capturing 1 and enable it

Initialize video capturing 2 and enable it

While True

Grab frame of camera 1

Grab frame of camera 2

If camera 1 then //Do Module 1

Initialize face detector model

Detection=net.forward() //detect faces through the model network

If length (detection) <=0 then //If no face detected

Continue Capturing

Else

Recognize Faces

If camera 2 then //Do Module2

Model the Background_Image

Grab the Current_Frame

Frame_difference= absdiff(Background_Image, Current_Frame)

If Frame_difference< Threshold then

Continue Capturing

Else

Classify Motion

If Motion class=Human then

Go to Module 1

End While

END

5.3 Face Recognition Implementation

The face recognition architecture in sub-section 4.2A of this document is a blueprint for the implementation of this module.

5.3.1 Face Detection Implementation

Input: Input video frames

Output: A set of bounding boxes that contain faces

Procedure:

This study employs a DNN pre-trained face detector model which is loaded from the disk using *cv2.dnn.readNetFromCaffe ()* function of OpenCV. Reading a network model stored in Caffe framework is with args for "prototxt " and "model" file paths as shown in the Figure below.

```
# load our serialized model from disk
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])
```

Figure 5. 1 Code Snippet for Loading Face Detector Model

The DNN network accepts a blob of an input image. So, *cv2.dnn.blobFromImage ()* function is utilized to constructs the blob of input image loaded with *cv2.imread ()* function. After this, the actual detection takes place by passing the blob through the network. With *setInput()* method of OpenCV, the new input value is initialized for the network which is the face blob and with *forward()* method the blob is passed to compute the output of layer.

```
# pass the blob through the network
net.setInput(blob)
detections = net.forward()
```

Figure 5. 2 Code Snippet for Face Detection

The final output of face detection implementation is a tight bounding box contains a face part. To get this output, we have to loop over the detection and draw a bounding rectangle with associated probability on it. This is done by utilizing *cv2.rectangle ()* and *cv2.putText ()* functions of OpenCV (see Appendix B1).

Collecting and labeling dataset

If not mentioned otherwise the face database for this study is prepared by the author programmatically by taking out the cropped face rectangle and store it on a disk. The process of data collection is depicted in Figure 5.3.

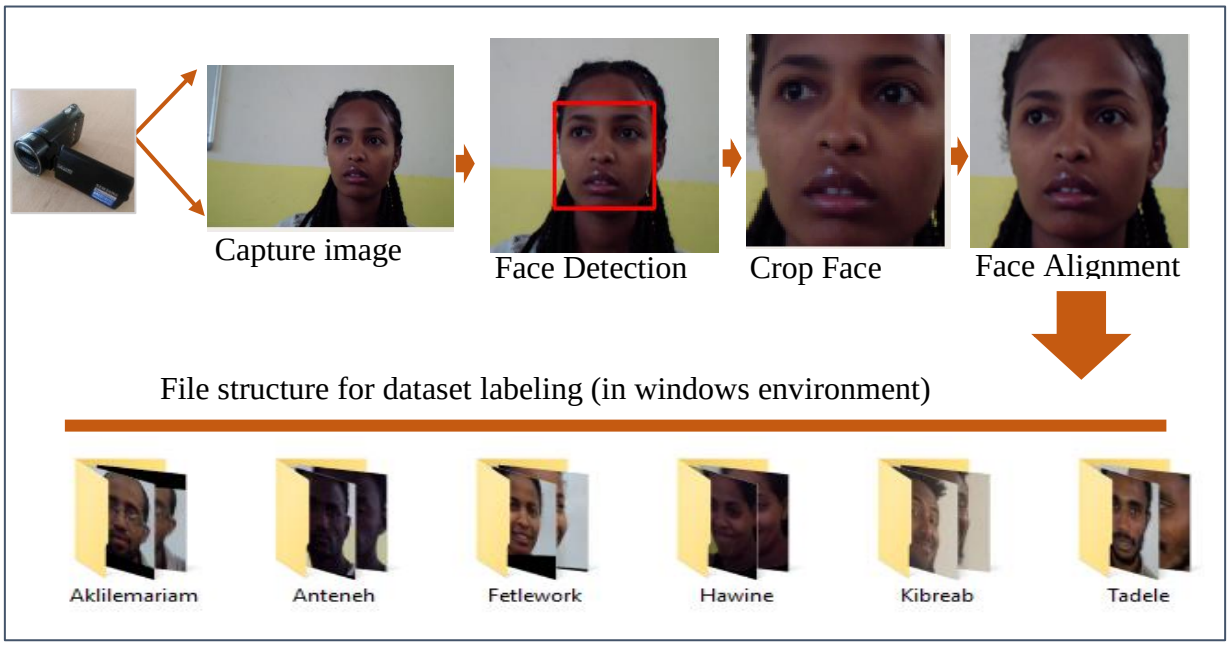


Figure 5. 3 Collecting and Labeling Face Dataset

Each folder shown in the Figure above represents different classes that contain a variety face images of individuals. The immediate crop of faces before the preprocessing step is given in the Figure below. Six faces in a block are from the same class contains the same person but in different challenges like variation in illumination, poses, and facial expression.

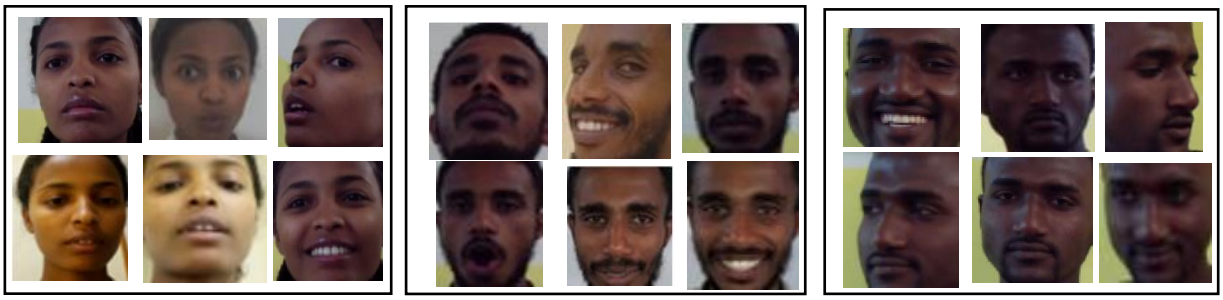


Figure 5. 4 Sample Face Images (Before Pre-processing)

5.3.2 Pre-processing Implementation

Face Alignment

The cropped faces shown in Figure 5.4 lose some detail information of facial points which affects the recognition accuracy. To compensate this, face alignment is done using a simple facial landmark to transform it into the output coordinate space.

A function *cv2.warpAffine()* creates a rotation matrix that can pack all the requirements of aligning (translation, scaling, and rotating). The outcome of this alignment process for the face images shown in Figure 5.4 is depicted as follows.

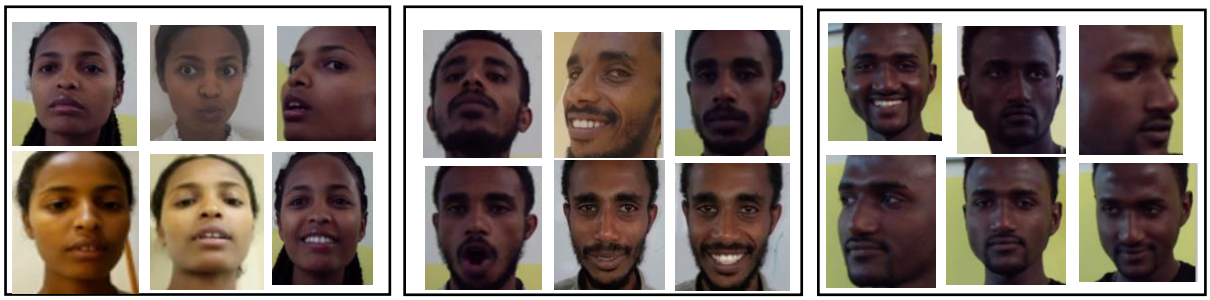


Figure 5. 5 Sample Face Images (After Alignment)

Gamma Correction

OpenCV provides an efficient way to perform gamma correction by building a lookup table that maps the input image pixel value to gamma-corrected value. This study used this efficient way depicted in Figure 5.6 to Gamma correct the dataset. The output looks like Figure 5.7.

```
#Gamma correctopn function
def adjust_gamma(image, gamma=1.0):
    # build a lookup table mapping the pixel values
    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
        for i in np.arange(0, 256)]).astype("uint8")
    # apply gamma correction using the lookup table
    return cv2.LUT(image, table)
```

Figure 5. 6 Code Snippet for Gamma Correction

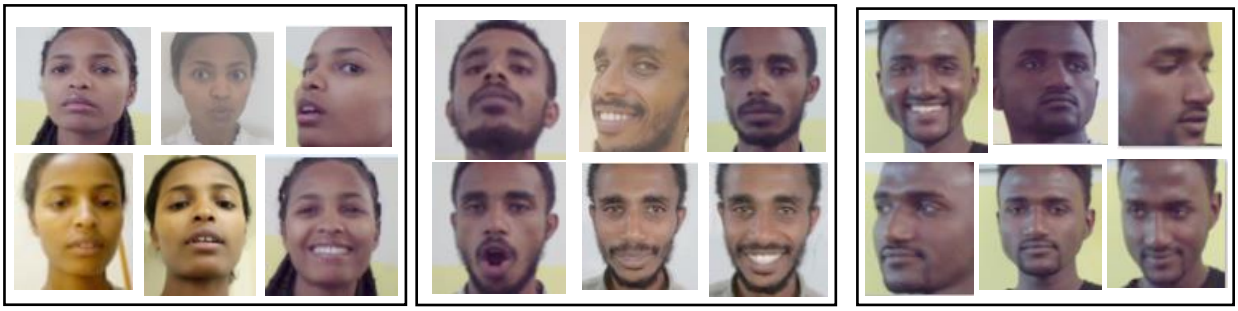


Figure 5. 7 Sample Face Images (After Gamma Correction)

Smoothing

The face dataset is taken under uncontrolled lighting conditions leads to some noisy images. To reduce this noise Gaussian Smoothing is employed by using *GaussianBlur()* function of OpenCV. Figure 5.8 shows a code snippet for this process followed by the resulting face images, Figure 5.9.

```
# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    fac = cv2.imread(imagePath)
    adjusted = adjust_gamma(fac, gamma=2.0)
    adjusted=cv2.GaussianBlur(adjusted, (13,13), cv2.BORDER_DEFAULT)
```

Figure 5. 8 Code Snippet for Face Image Smoothing

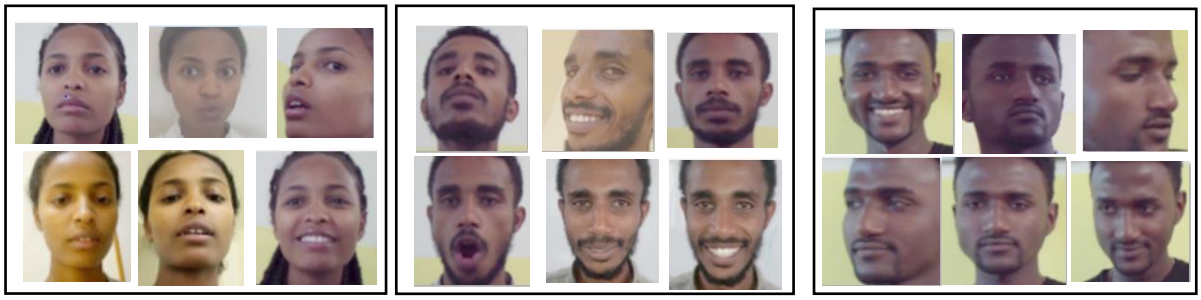


Figure 5. 9 Sample Face Images (After Smoothing)

5.3.3 Feature Extraction Implementation

Input: A cropped and enhanced face images obtained in the above two subsections

Output: 128-D feature vector known as face embedding

Procedure:

The face embedding model utilized in this study accepts a blob of face ROI, which is obtained during face detection implementation and stored in a face database. After reading this face image the blob is constructed using *cv2.dnn.blobFromImage()* function. The face blob is passed through the model to get a 128-D feature vector (see Figure 5.10).

```
# Face blob passing through the embedding model
faceBlob = cv2.dnn.blobFromImage(adjusted, 1.0 / 255,
    (96, 96), (0, 0, 0), swapRB=True, crop=False)
embedder.setInput(faceBlob)
vec = embedder.forward()
```

Figure 5. 10 Code Snippet for Feature Extraction

The feature vector extracted is dumped to a pickle file for future use as depicted in Figure 5.11. This is started with initializing an empty list for a feature and corresponding class name as feature = [] and labels = [].

```
# add the class name + face embedding to their respective lists
feature.append(name)
labels.append(vec.flatten())
# dump the facial embeddings + names to disk
print("[INFO] serializing {} encodings...".format(total))
data = {"embeddings": feature, "names": labels}
f = open(args["embeddings"], "wb")
f.write(pickle.dumps(data))
f.close()
```

Figure 5. 11 Code Snippet for Writing Feature Vector on a Pickle File

5.3.4 Face Classification Implementation

Training a Model

Loading a feature vector dumped on a pickle file in section *5.3.3* a Logistic Regression classifier is learned as illustrated with the code snippet 5.12.

```
# load the face embeddings
data = pickle.loads(open(args["embeddings"], "rb").read())
#Encode the labels
le = LabelEncoder()
labels = le.fit_transform(data["names"])
(trainX, trainY) = (data["embeddings"], labels)
Kf=KFold(n_splits=10, random_state=42, shuffle=False)
for train_index, test_index in Kf.split(trainX):
    (x_train, x_test, y_train, y_test)=trainX[train_index],
    trainX[test_index], trainY[train_index], trainY[test_index]
    recognizer=LogisticRegression(solver="lbfgs", multi_class="multinomial")
    recognizer.fit(x_train, y_train)
    recognizer.score(x_train, y_train)
```

Figure 5. 12 Code Snippet for Training a Face Recognition Model

Testing a Model

To evaluate the model skill a model is fitted on a test set and the result of prediction is visualized in the form of a confusion matrix (see Figure 5.13).

```
# evaluate the model
preds = recognizer.predict(x_test)
print(confusion_matrix(y_test, preds))
print(classification_report(y_test, preds, target_names=le.classes_))
result = recognizer.score(x_test, y_test)
print("Test_set classification accuracy:",result)
```

Figure 5. 13 Code Snippet for Testing Face Recognition Model

The face recognition model is also tested from a real-time test video. The sample python code to do this is presented in Appendix B2.

5.4 Human motion Detection Implementation

The human motion detection architecture depicted in sub-section 4.2B of this document served as a blueprint for the implementation of this module.

5.4.1 Motion Segmentation Implementation

Input: Video Frames

Output: A bounding box that contains a motion part of the input frames

Procedure:

This study used a Mixture of Gaussian (MOG) based Adaptive Background Subtraction method to detect the part of an input video contains a motion. To get the difference between the background model and a current frame a function called *BackgroundSubtractorMOG2()* is used. This subtractor is applied to each and every input frame updating a background using 60 previous frames. On the result of the frame difference image, a contour is detected on which a bounding box is drawn using *cv2.rectangle()* function (See Appendix B3).

Using the motion segmentation implementation the human dataset employed for motion classification is collected. Figure 5.14 below shows this process. The Adaptive Background Subtraction stage generates an approximate bounding box enclosing the object in motion. The bounding box is then resized to a dimension of 64X128 pixels and stored in a disk as a positive dataset (Human Dataset).

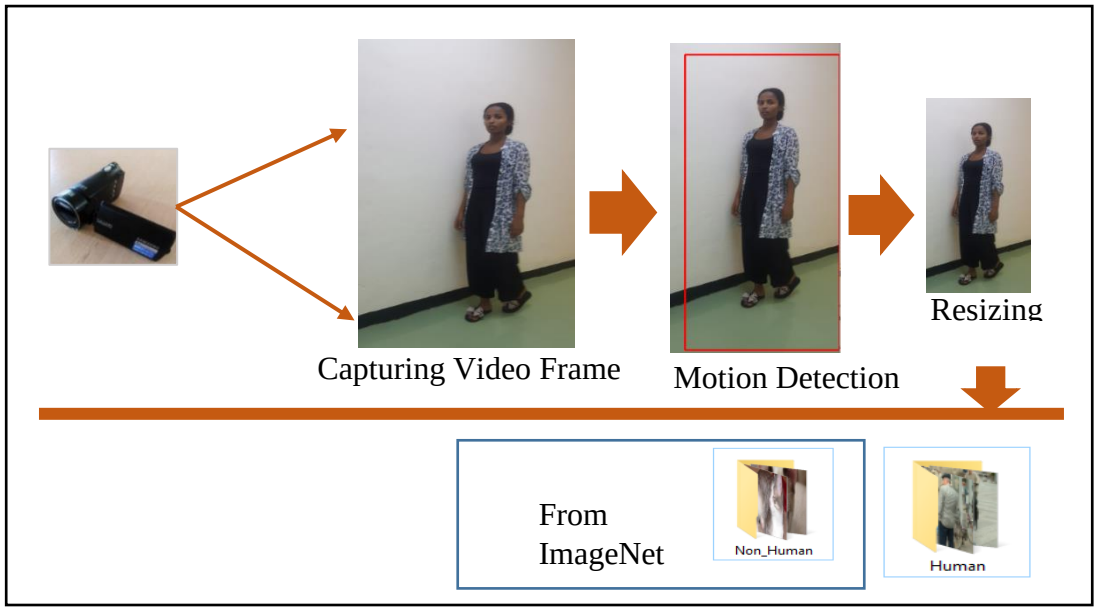


Figure 5. 14 Collecting and Labeling Human Dataset

Motion recognition as a binary classification problem needs a positive dataset, collected and stored like the above Figure, and a negative dataset that could contain any image but not the object of interest (Human). However, it is a good practice to select the negative dataset based on the nature of the problem in hand. In the case of this study, non-human objects that can move in an indoor environment are mostly pets. Therefore, pet images are collected from one of the biggest computer vision online databases ImageNet.

5.4.2 Pre-processing Implementation

The bounding box obtained during motion segmentation has a variant size because moving objects have different sizes and height. To make the system invariant to these difference the box is resized to 64X128 dimension using *cv2.resize*() function. In addition, false positives because of illumination and shadow effect during the motion are cleaned up using morphological operations shown in the following code snippet. The result can be traced to Figure 5.16.

```
# Enhancement of Motion ROI
thresh = cv2.threshold(fgmask, 50, 255, cv2.THRESH_BINARY)[1]
kernel=np.ones((5,5), np.uint8)
erosion=cv2.erode(thresh, kernel, iterations=1)
dialation = cv2.dilate(erosion, kernel, iterations=1)
opening=cv2.morphologyEx(dialation, cv2.MORPH_OPEN, None)
closing=cv2.morphologyEx(opening, cv2.MORPH_CLOSE, None)
```

Figure 5. 15 Code Snippet For Pre-processing Motion Image

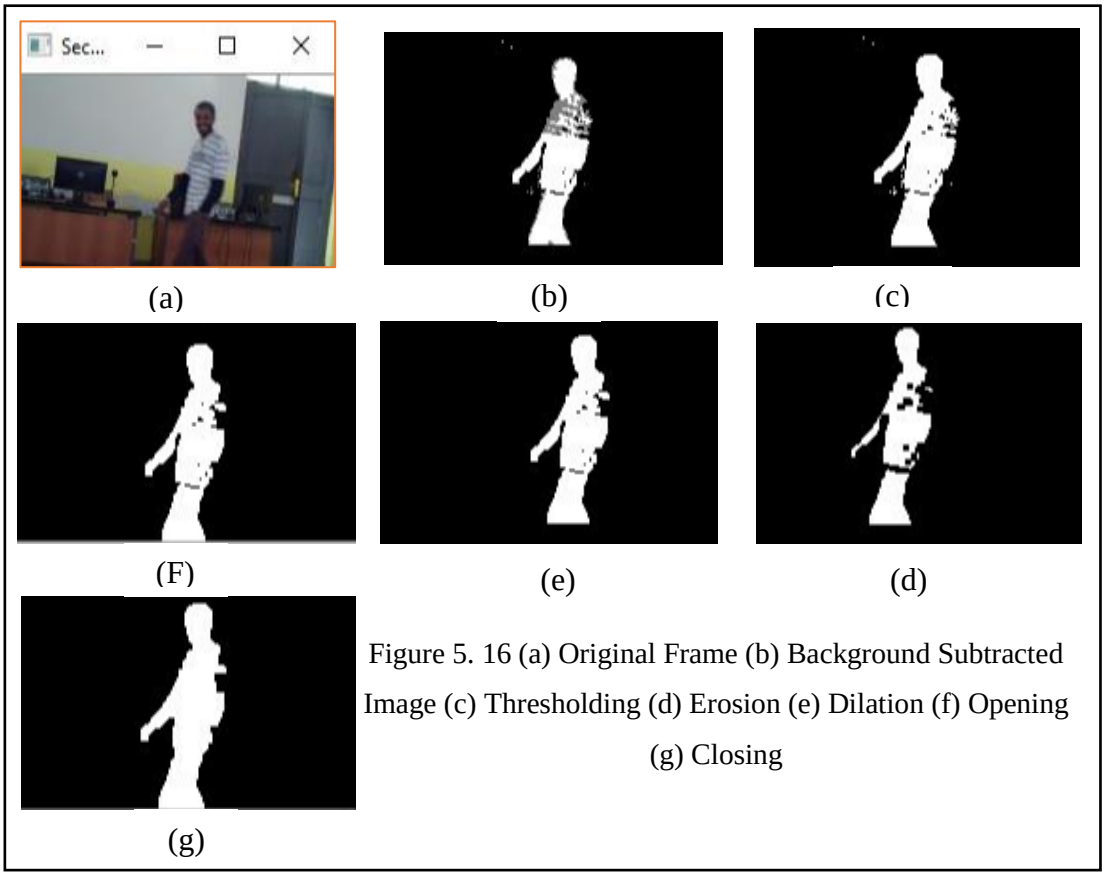


Figure 5. 16 (a) Original Frame (b) Background Subtracted Image (c) Thresholding (d) Erosion (e) Dilation (f) Opening (g) Closing

Image (b) in the above Figure shows the result of an Adaptive Background Subtraction method which contains a foreground object with many noises. To remove those noisy spots first thresholding is applied as image (c). Erosion, image(d) creates a continuation of small black boxes (noises) on the biggest white blob which is then filled by dilation image (e). After that opening and closing remove false negative and false positive respectively.

5.4.3 Feature Extraction Implementation

Input: A cropped and enhanced motion region obtained in the above subsections

Output: HOG Feature Descriptor

Procedure:

The motion region is divided into multiple regions of the fixed rectangular size called cells. The gradient is calculated for each cell and weighted by the strongest gradient. Then the HOG descriptor is applied using orientation=9 for every 4X4 pixels-per-cell as depicted in the code snippet below.

```
# Histogram of Oriented Gradients descriptor from the ROI
H = feature.hog(entity, orientations=9, pixels_per_cell=(4, 4),
    cells_per_block=(2, 2), transform_sqrt=True, block_norm="L1")
```

Figure 5. 17 Code Snippet for HOG Feature Extraction

5.4.4 Motion Classification Implementation

Training a model

The database is split into training and testing set and a linear SVC from class sklearn.svm is trained as a motion classifier.

```
# Training a motion classifier
(trainX, trainY) = (data["features"], labels)
(x_train, x_test, y_train, y_test) = train_test_split(trainX, trainY, test_size = 0.10)
recognizer= SVC(C=1.0, kernel="linear", probability=True)
recognizer.fit(x_train, y_train)
recognizer.score(x_train, y_train)
```

Figure 5. 18 Code Snippet for Training Motion Classifier

Testing Model

The learned model is fitted on a testing data to evaluate how well it can make a prediction on previously unseen data. The result was visualized in the form of a confusion matrix with a classification report of accuracy (see the code snippet below).

```
print("[INFO] ...evaluating the motion classifier...")
preds = recognizer.predict(x_test)
print(confusion_matrix(y_test, preds))
print(classification_report(y_test, preds, target_names=le.classes_))
result = recognizer.score(x_test, y_test)
print("Test_set classification accuracy:",result)
```

Figure 5. 19 Code Snippet for Testing Motion Classification Model

The human motion classification system is also tested from a real-time test video. The sample python code to do this is presented in Appendix B3.

CHAPTER 6: RESULT AND DISCUSSION

6.1 Overview

This Chapter reports the overall result of the study with the interpretation of the result. The experimental result of both modules is measured and discussed separately with the help of accuracy metrics. The measurements are then evaluated by comparing the result with the previous literatures and other competitive methods. The primary objective of the comparative analysis is to evaluate the reliability of the methodological approach used to implement the proposed architecture. On top of the accuracy metrics, the designed architecture is evaluated using a scenario-based evaluation approach. This is to ascertain whether or not the proposed integrated architecture is robust as expected. A research result should answer the research questions in hand [47]. Therefore, each result in this thesis is organized logically alongside the research questions stated in Chapter One.

6.2 Robustness and Reliability of the Current Vision-based Architecture for Indoor Environment Security

The security system as a crucial aspect of human life puts a greater demand on software quality. Even if the software quality is the focus of many researches the main quality attribute of interest is dependent on the problem under investigation. Having the aim of improving the reliability of an indoor environment security system this study focuses on the accuracy and robustness of the system, a sub-category of reliability.

The robustness of the current home security architecture is answered by the literature review section of this thesis. Robustness is the degree to which a system or component can operate correctly in any possible operational conditions [48]. According to the result extracted from subsection 2.4 of this document the current architecture failed to operate correctly for at least two operational conditions (scenarios 1 and 2). This shows that the currently existing vision-based indoor security systems are not robust and therefore not reliable enough.

6.3 Robustness and Reliability of the Proposed Integrated Architecture

The vulnerability of a security system should be evaluated with a direct interaction between man and machine because a condition to forge a system is created by intruders. Therefore, this study formulates five different scenarios to answer **RQ2** stated in section 1.4 by examining the system performance against both usual and unusual activities.

Scenario 1: Authorized person arrives at the entrance door

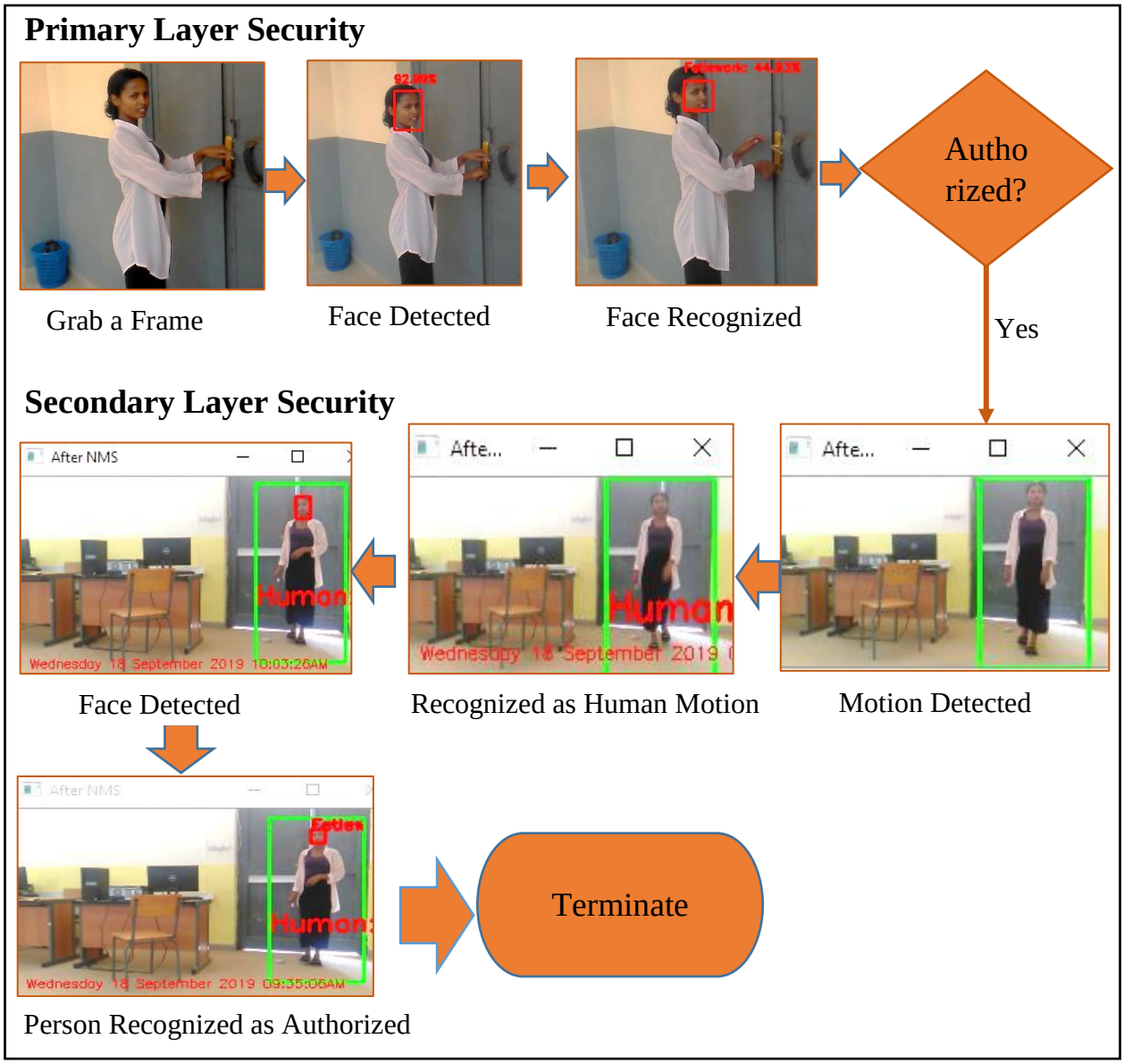


Figure 6. 1 System Evaluation Result:Scenario 1

The Figure above illustrates the interaction between the proposed system and a legitimate user who has access to the restricted indoor environment. The result obtained confirms that the proposed system works correctly as expected. The architecture proposed in paper[33], triggers a false alarm in this scenario because as soon as motion is recognized as human it activates the alarm system even for authorized persons.

Scenario 2: Unauthorized person arrives at the entrance door

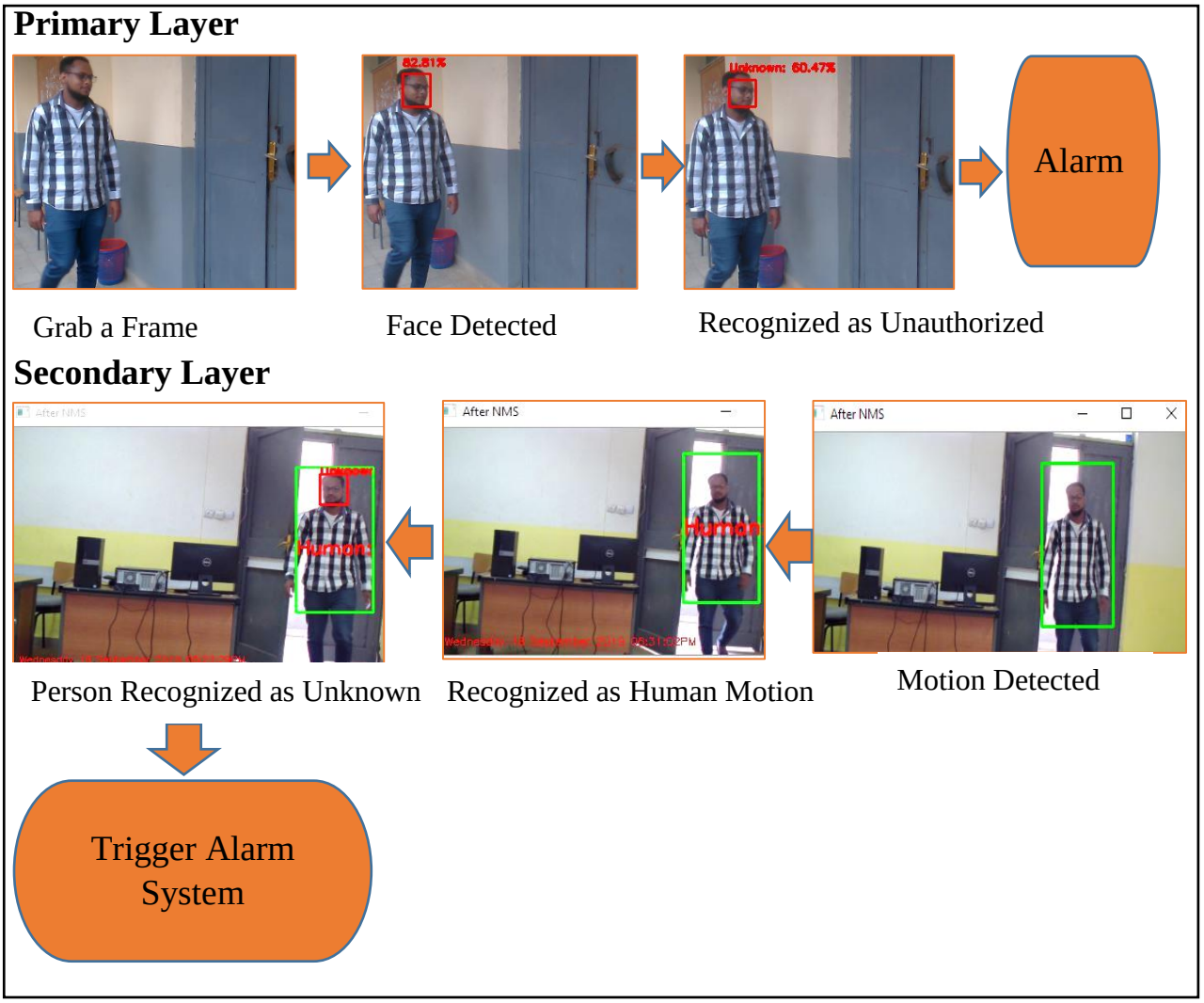


Figure 6. 2 System Evaluation Result:Scenario 2

In the case of unauthorized persons who are not allowed to enter into the surveilled indoor environment, the primary layer security triggers security alarm as shown in Figure 6.2. This is a good advantage to deter burglary before it actually occurred since the alarm can make an intruder run away instead of entering the room. Even if an intruder gets inside ignoring the ringing alarm, which is a rare case, he/she would be under control of the secondary layer security system as shown in Figure 6.2.

Scenario 3: Burglar wearing a mask arrives at the entrance door

Primary Layer Security

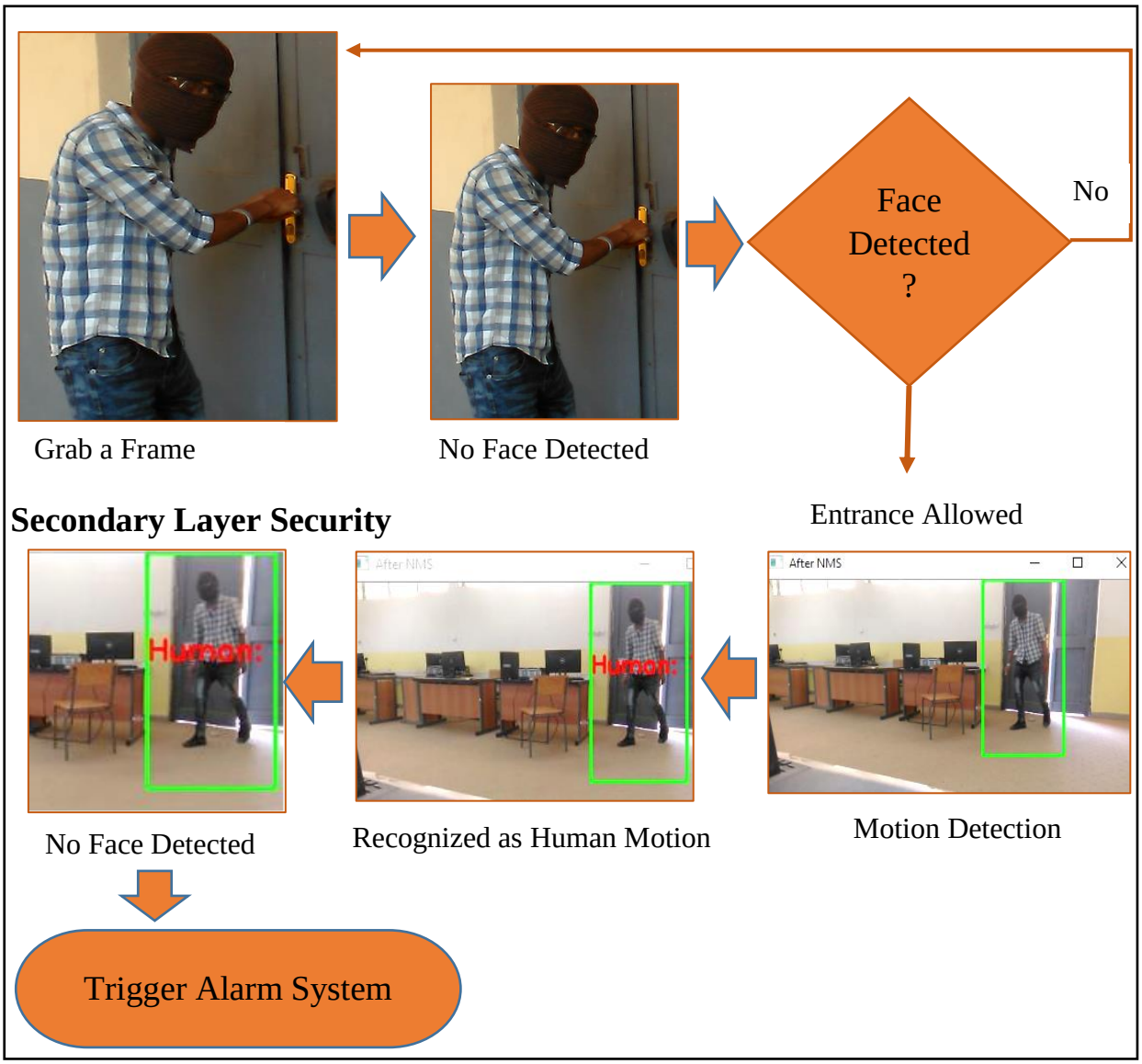


Figure 6. 3 System Evaluation Result: Scenario 3

The very challenging condition in a fully vision-based home security system is when intruders come with a mask as shown in the above scenario. This full occlusion of the face deceives the primary layer not to trigger the security alarm and the burglar enters the room without any trouble. However, we don't have to worry because the secondary layer security can take care of this situation. It activates a security alarm recognizing that there is definitely a human intrusion but might be with an occluded face.

Scenario 4: Burglar gets inside the room through other means

A: If the burglar doesn't wear a mask

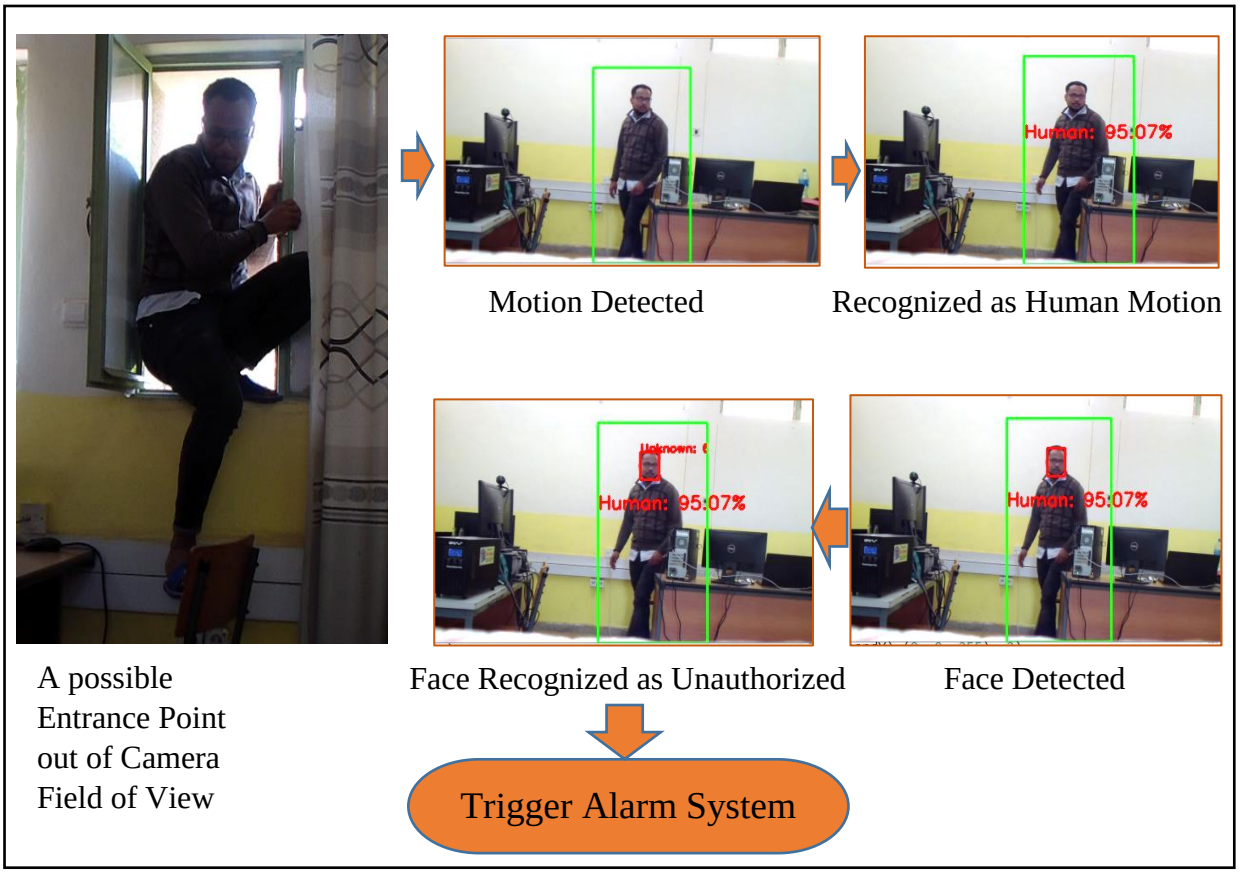


Figure 6. 4 System Evaluation Result: Scenario 4A

B: If the burglar wears a mask

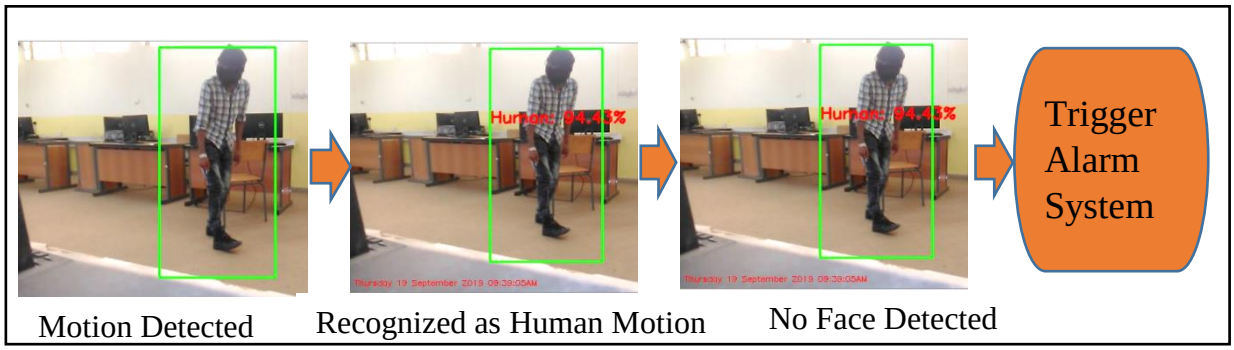


Figure 6. 5 System Evaluation Result: Scenario 4B

Figures 6.4 and 6.5 show the ability of the system to guard the restricted area when a burglar uses an entrance point other than the front door. The currently available vision-based architecture completely fails to provide the required functionality in this situation as clearly stated in section 2.4 of this document.

Scenario 5: Home owner's pet walking around the room

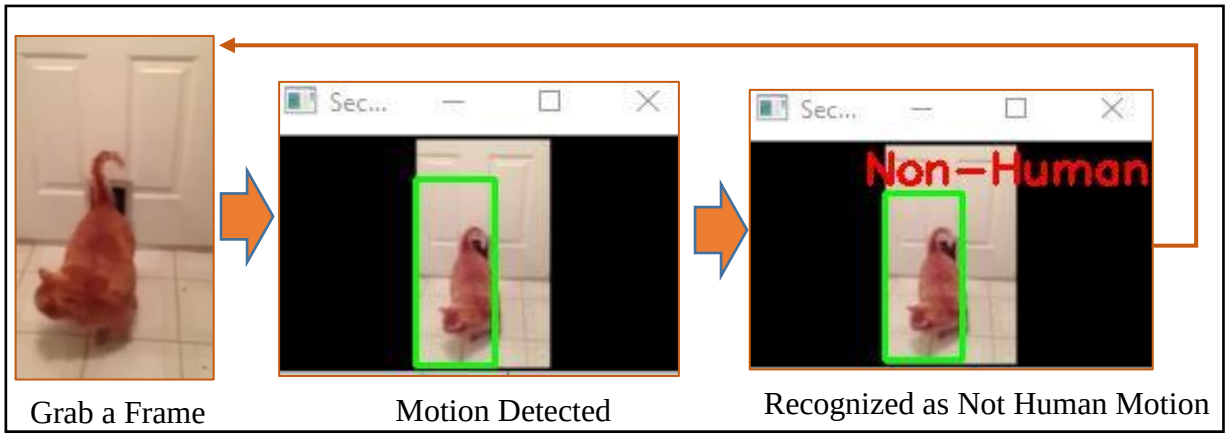


Figure 6. 6 System Evaluation Result: Scenario 5

The single-layered home security systems based on motion detection [27], produces an annoying false alarm due to the motion of non-human moving objects, most probably pets. This is because such systems start buzzing alarm whenever motion is detected. From the result obtained in Figure 6.6, the proposed system avoids such false alarm by doing a further classification of the motion area as human or not. In the above scenario, the system recognizes

the motion is caused by this lovely cat in the picture and therefore continues to grab the next new frame.

6.4 Robustness of the Methods Applied to Implement the Proposed Architecture

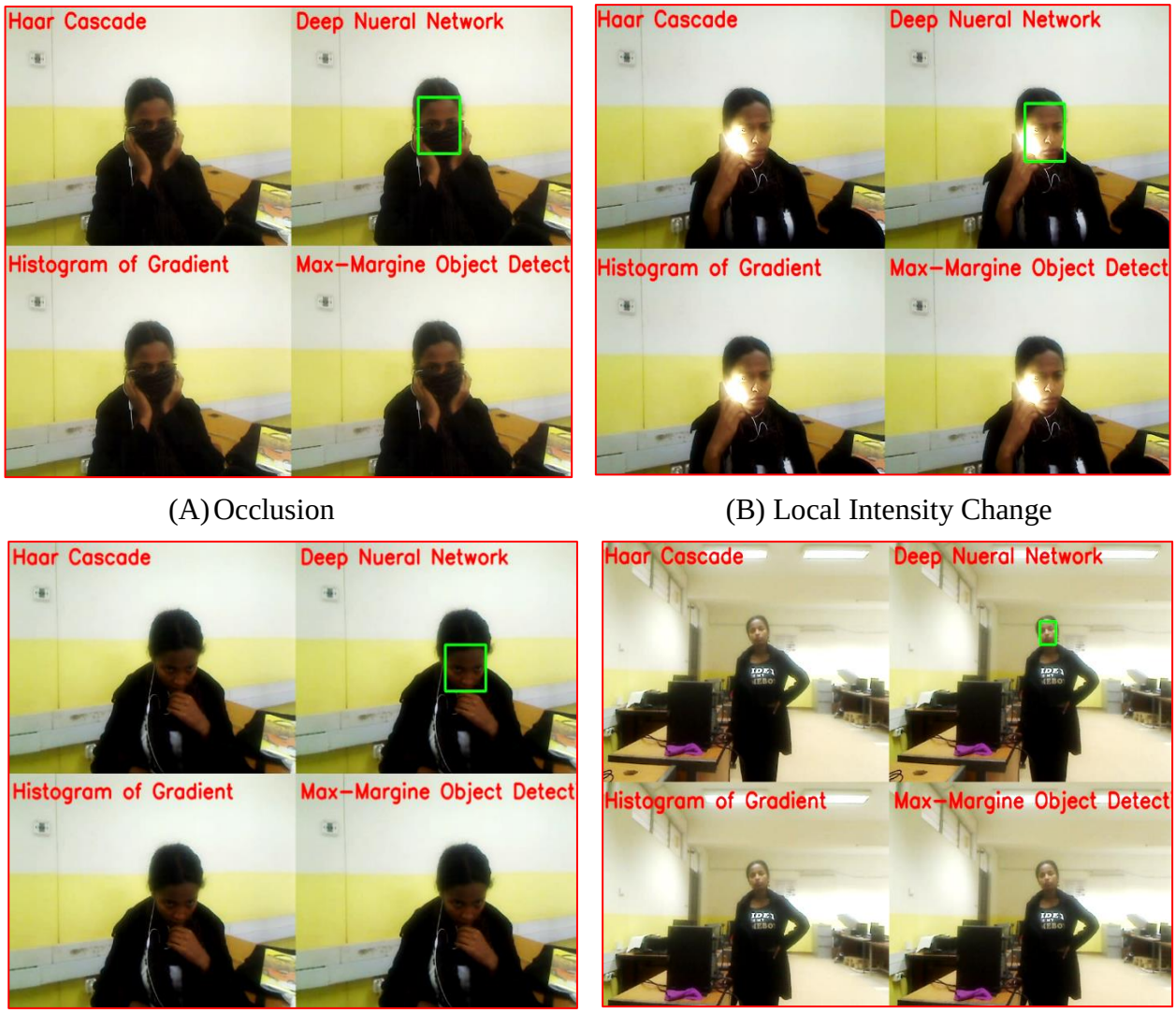
Previous researchers used their own methodological approach to develop vision-based in-house security systems. This study followed a different method combination to implement the proposed architecture. To evaluate the robustness of those methods various comparative experiments have been conducted at each step of the system development (**RQ3**).

6.4.1 Face Detection

This section discusses the comparative result of the face detection experiment. The input is a live video from the webcam with different face detection challenges like partial occlusion, bad pose, bad illumination, and variable distance from the camera. From the Literature Review section of this document, it is understandable that most of the previously done related works used a Viola-Jones (Haar Cascade) face detector. Figure 6.7 below evaluates the efficiency of the method employed in this work comparing with Viola-Jones and the other two most commonly used face detectors.

The big problem of Viola-Jones face detector is it can only detect a face that is upright frontal to the camera. In addition, the Haar-Cascade detector is very sensitive to local illumination change and partial occlusion. Such methods are not efficient especially for surveillance systems because intruders are non-cooperative subjects who try to disguise themselves from biometric authentication in any possible way.

From the result illustrated in Figure 6.7, it can be seen that the DNN face detector employed in this thesis outperformed all other methods. It works for different face orientations, under substantial occlusion across various scales. To partly answer the **RQ3** in Sub-section *1.4* of this document the face detection method for surveillance systems should be such a resilient to different challenging factors.



(A) Occlusion

(B) Local Intensity Change

(C) Bad Pose

(D) Long Distance

Figure 6. 7 A Comparative Result of Face Detection Methods

The face detection problem such as partial occlusion, bad pose, and lighting conditions are getting improvement with the advent of Deep Neural Network. In DNN the machine is learned with plenty of face images with strong challenging factors. Therefore, they are comparatively robust techniques. However, long-distance face detection is a long-lasting challenge. From the experiment result obtained even the DNN face detector which is robust for other challenges detect a face if only the distance between a person and a camera is up to 7 meter. Therefore, it is recommended to work on long-distance face detection for surveillance applications.

6.4.2 Face Recognition

This section discusses the comparative result of the face recognition experiments. The dataset used to implement and test the face recognizer is made using 600 face images, 100 images per each of six classes. All images are in 256x256 resolution, with .jpg extension. Each face image in the database is taken with variation in orientation (tilt to different directions), illumination (dark/bright), facial expression (close/open eyes, smiling, sad, etc.), facial details (glass/ no glass) as shown in Appendix A1. In the experiment, the entire dataset is randomly split into 10 folds to apply a 10-fold cross-validation evaluation technique. The plot of the resulting confusion matrix in Figure 6.8 shows the performance of the implemented face recognition system.

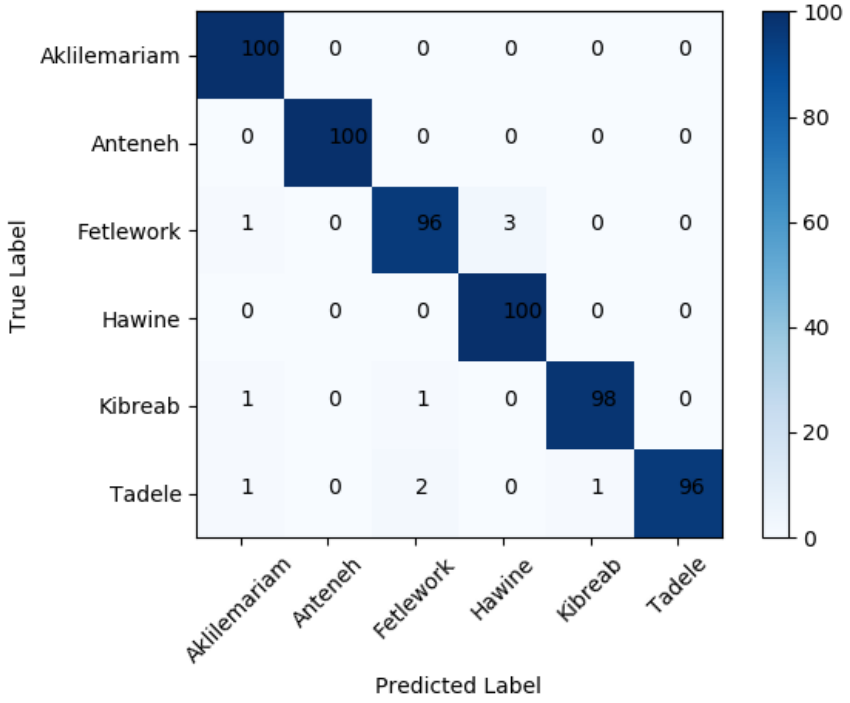


Figure 6. 8 Face Recognition Result**:** Recognition Performance

This study used a novel combination of FaceNet feature extractor with Logistic Regression classifier to recognized faces and achieved a good recognition rate of 98.3%. However, there are many other deep learning architectures and well-known classifiers that can be employed for face recognition problems. This study conducted various experiments to compare the proposed

face recognition model with 14 other models. The result obtained is presented in Table 6.1 below.

Table 6. 1 Performance Comparision between Face Recognition Models

Random Split (70:30)	Feature Extractor/Classifier	FaceNet	ResNet50	VGG16	VGG19	Xception
	Logistic Regression	95.0%	89.9%	64.0%	93.0%	79.8%
	SVM	26.4%	91.1%	25.9%	26.5%	19.2%
	Random Forest	92.9%	98.0%	91.0%	80.2%	87.8%
10-Fold Cross-Validation	Feature Extractor/Classifier	FaceNet	ResNet50	VGG16	VGG19	Xception
	Logistic Regression	**98.3%**	96.2%	89.0%	96.5%	82.8%
	SVM	39.0%	95.8%	27.6%	42.0%	19.0%
	Random Forest	97.5%	95.8%	92.8%	95.3%	87.0%

On top of the highest accuracy shown in the table above the other reason to choose FaceNet instead of other deep learning architecture is its representational efficiency. In contrast to other approaches, FaceNet directly trains its output to be a 128-D compact vector. This is of great advantage to employ it on a resource-limited device and to decrease the time required to extract feature embedding. During the experiment of this study, the time taken to extract features from a given face image is recorded and compared in Table 6.2.

Table 6. 2 Representational Efficiency of Feature Extraction Methods

CNN Architecture	Feature Dimension Per Face	Feature Extraction Time in Minute
FaceNet	128	0.025
VGG16	25,088	0.13
VGG19	25,088	0.16
ResNet50	100,352	0.23
Xception	100,352	0.26

In terms of representational efficiency and response time, the proposed algorithm is far superior compared to all other algorithms in Table 6.2. For surveillance systems, feature extraction methods should be such efficient to represent an image and fast to give a response. This is because security systems are real-time applications that are mostly embedded in resource-limited devices.

Performance comparison was conducted by implementing the proposed face recognition method with different dataset sizes. There are six variations of data size used, i.e 50,75,100,125, 150 and 175 face images per each class. The following graph represents the pattern of accuracy obtained for different dataset sizes.

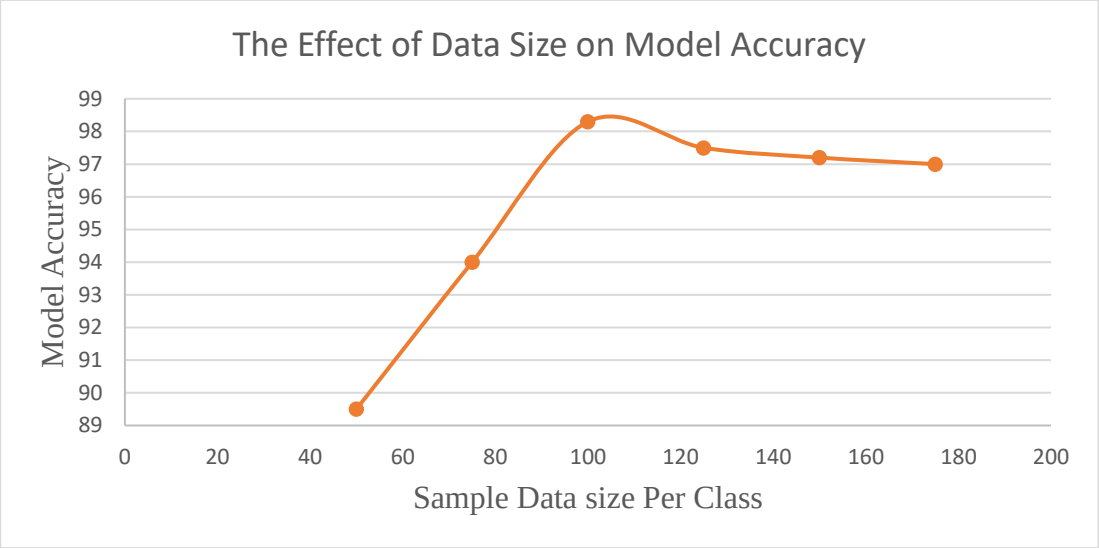


Figure 6. 9 Face Recognition Performance Result**:** Different Data Size

Based on the experiment result presented in the above Figure, different sized training datasets yield different model performance. The accuracy is low for the small training set, perhaps the samples are not sufficiently representative of the class. The pattern shows that the accuracy is increased with the increased dataset size up to 100 and slightly lower for the rest three data sizes. The large size of the dataset lowered the accuracy, even slightly, perhaps the model doesn't have a capacity to learn the fine detail of such large samples.

The model skill comparison is also conducted by varying the number of classes in a face recognition problem. A number of classes starting from two up to six was the variation

considered. The resulting recognition rate for each corresponding class size is plotted as the following graph.

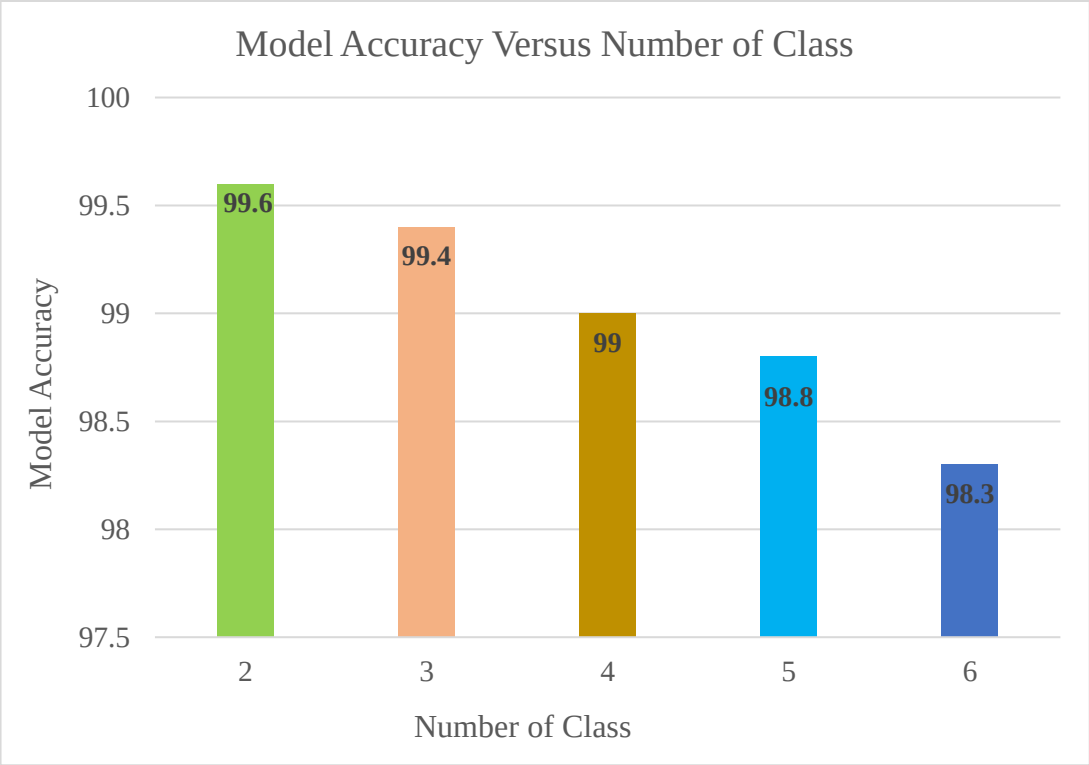


Figure 6. 10 Face Recognition Performance Result**:** Different Class Sizes

The experiment indicates that the performance of the model decreases as the class size increases (see Figure 6.10). This is because as the number of classes grows the precision of classification per class can reduce as a given image has the highest probability to be misclassified. Therefore, in the case of application areas where the number of classes is very large, it is recommended to use a lot of training examples for each class. This is to make more accurate feature selection that can improve the precision of classification.

The performance of a classification model is highly affected by the pre-processing stages performed on the dataset in hand. This study conducted experimental analysis on the performance of the face recognition model using a dataset before pre-processing and after pre-processing using the appropriate technique chosen. The result is reported in Figure 6.11 below.

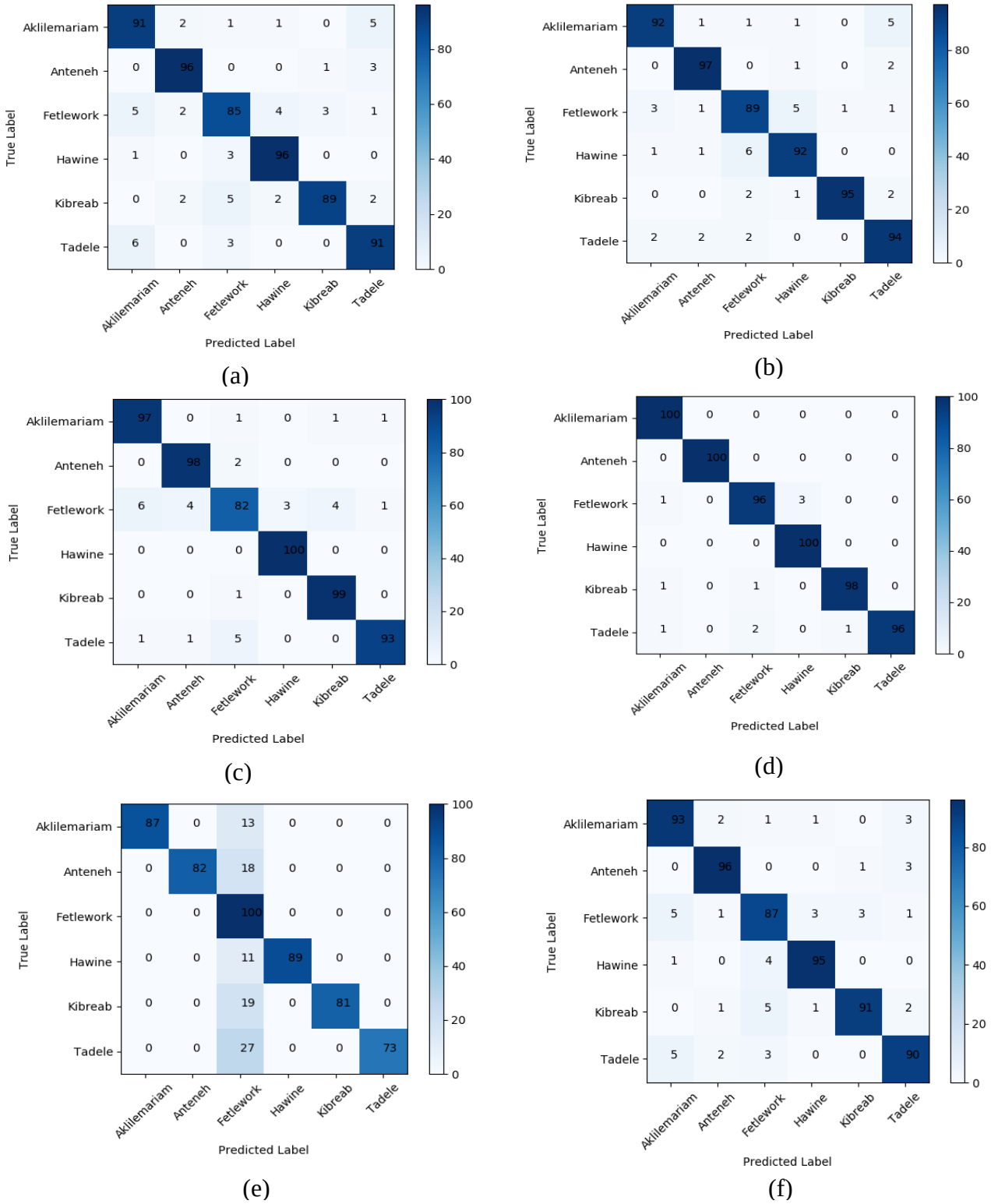


Figure 6. 11 Face Recognition Result**:** (a) Before Pre-processing (b) After Face-Alignment (c) After Gamma-Correction (d) Gamma-Correction followed by Smoothing (e) After Smoothing (f) Smoothing followed by Gamma-Correction

Based on the above result, the accuracy of recognition before doing any preprocessing is 91.33 % (Figure 6.11a). Which is improved to 93.16% after applying face alignment and 94.83% after doing a Gamma-Correction on the training dataset. The final result presented in Figure 6.11d is even more accurate at 98.3%, which is obtained after Gamma-Correction followed by Smoothing (Gaussian-Blur). The result of performance after smoothing (c) is 85.33% and (d) resulted in 92%. This low accuracy is perhaps because smoothing on the low contrast image region makes a distinctive feature of a given image, such as edges, to be diminished. The following graph represents the effect of pre-processing on the face recognition model accuracy.

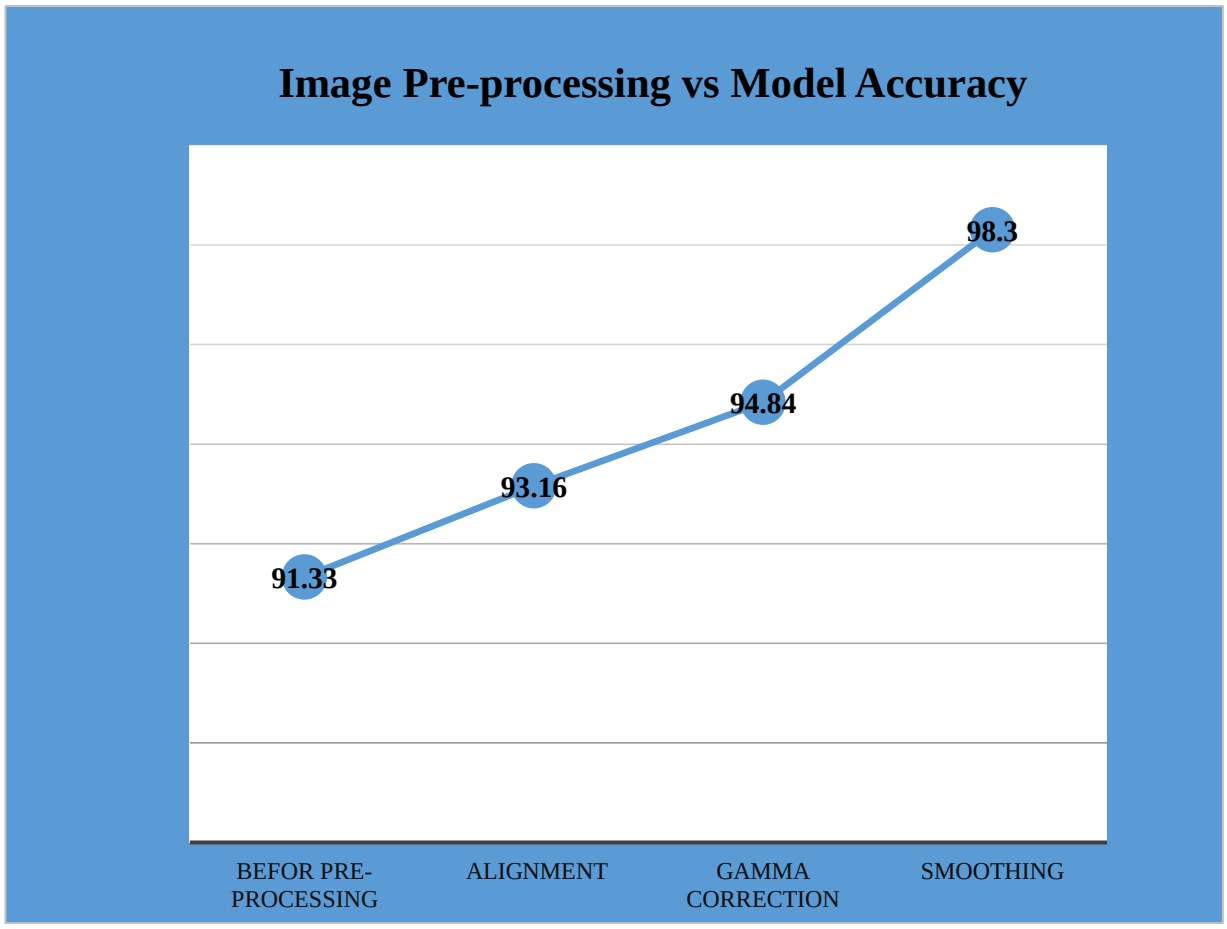


Figure 6. 12 Face Recognition Performance Result: Pre-processing Effect

6.4.3 Motion Segmentation

This section discusses the comparative result of the motion segmentation (detection) experiments. The input is a live video from the webcam with different challenges like a variation in frame rate, illumination, and noises. The most commonly used methods in related studies

have tested with the same video input to evaluate the efficiency of the proposed segmentation technique. Simple BGS and AFS method doesn't extract the silhouette of a moving object correctly when the object moves slowly and smoothly where there is relatively small motion in the motion area due to the frame differencing. Figure 6.13 below shows the result of these two methods compared with the proposed adaptive technique.

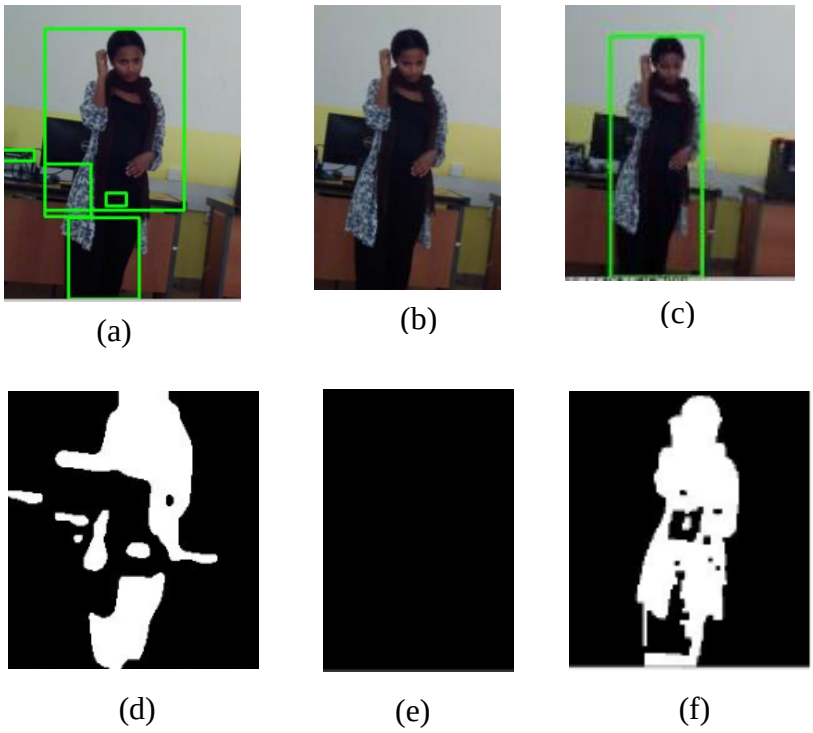


Figure 6. 13 Comparision of different motion segmentation methods with bounding box on the motion region (a) AFS (b) BGS (c) Adaptive Background Subtraction and (d) –(f) is the corresponding silhouette

Ground-truth experiments with an input video have shown that the AFS method can detect the foreground mask but many background pixels are incorrectly classified as foreground pixels, small rectangles in Figure 6.13 (a). This is caused by non-stationary background objects such as shadows cast by the moving object or instant lighting conditions. From the result, the ghost mask image of AFS (d) is not extracted accurately and therefore it would be difficult for further motion recognition step.

On the other hand, the BGS method results in false-negative as shown in Image (b) and (d), this is perhaps because of the high frame rate of the sequence with a slowly moving object, 30 fps speed. The adaptive background subtractor shown in the image (c) performs better. Most

importantly the relatively accurate motion region helps the next motion classification system to perform well. This is because this method automatically adapts to the environment in every frame.

There are different parameters that affect the output of adaptive motion segmentation methods such as length of history (frames to adapt to the environment) and a threshold value. To boost its performance under the give environmental conditions various experiments of parameter tuning have been performed. The good result is obtained with a History of 60 and VarThreshold of 50 pixels. From the parameter tuning experiment of these parameters, an impact on the result of foreground extraction can be summarized as the table below.

Table 6. 3 Motion Segmentation Result Comparision: Different parameters

Parameter	Amount	Result
History	Very Small	The slow adapting model can't quickly overcome large background changes. This results in many false positives and a ghost mask trails the actual object.
	Very Large	The Fast adapting model fails at low frame rates, susceptible to noise and aperture problem.
VarThreshold	Very Small	The foreground extraction is extremely inclusive, with large false positives.
	Very Large	The foreground extraction ends up with many false Negatives.

The summary in the above table concludes that the parameter should be carefully chosen by experimenting and examining different factors.

6.4.4 Motion Recognition

This section discusses the result of the motion classification. The dataset used to implement and test the motion recognizer is made up of 400 images, 200 images per each of two classes. All images are in 64X128 resolution, with .jpg extension. Each image in the positive database is taken with variation in walking style and illumination (dark/bright). The sample dataset used for human motion classification is shown in Appendix A2. The motion classification result is highly affected by the input feature vector extracted using the HOG method and the performance of the

classifier. To pick point the suitable parameter that gives a distinctive feature descriptor different experiments were done on the key parameter of the HOG method. Based on the experiment the key parameters are decided to be :

```
Input size= (200, 100)
orientations=9
pixels_per_cell=(4, 4)
cells_per_block=(2, 2)
```

With the HOG vector resulted using the above parameters and SVM classifier the motion region classification process is 90% accurate. The accuracy achieved is illustrated with the confusion matrix in Figure 6.14 below.

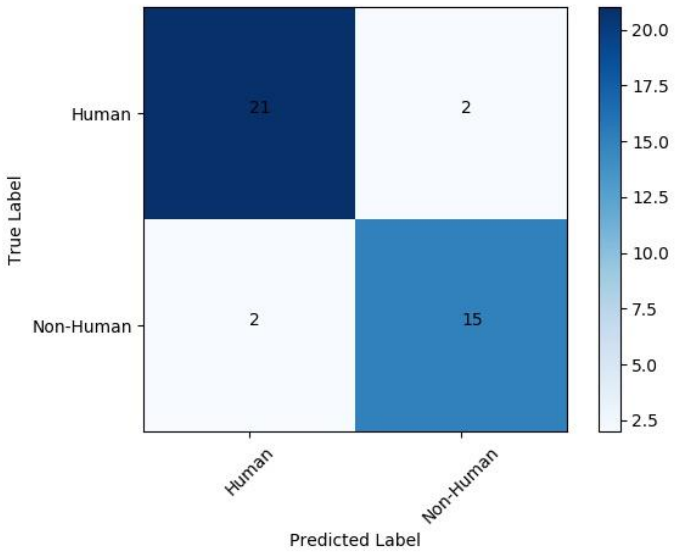


Figure 6. 14 Motion Recognition Result: Recognition Performance

Even if it is better than the accuracy of motion classification claimed by previous researchers 90% accuracy is not enough. Therefore, this study encourages researchers to contribute to improving this result.

This study also conducted a field test to evaluate how much the proposed system is responsive to real-time applications and it achieved an acceptable computation time result. In a real-life scenario, the system takes about 1.505 seconds to recognize individuals and 0.005 seconds to

identify a motion class. The result is affected by the performance limitation of the hardware device used. It can be highly improved by using a high-performance machine.

6.5 Advantages and Disadvantage of the Proposed Architecture

The Proposed Architecture has the following merits:

- Prevent the robbery before it actually occurred since the alarm at the door makes the intruder run away without breaking into the room.
- Reduce annoying false alarms caused by moving non-human objects
- More robust and reliable to resist burglar activity of forging a security system
- It provides an essential base to develop more complicated vision-based security systems with which it can be integrated possibly easily
- The fully vision-based nature of the architecture characterizes it to be affordable, unobtrusive, simple and usable.

The proposed architecture has its own demerits stated below:

- It requires a high-performance server machine to get a good real-time response
- It can reduce but can not totally avoid false alarms, the subsystems might be deceived through a spoofing attack, this can be solved by adding an additional layer that can analyze the intruder activity (Unusual Activity Analysis).

CHAPTER 7: CONCLUSION AND FUTURE WORKS

This chapter concludes the research described in this thesis. The aim and objective of the research, outlined in Chapter One, are reviewed to address their achievement. The main contributions of the dissertation are clearly stated and at the end, there are prospective points indicated by this research that will be done in the future.

7.1 Conclusion

Nowadays, automated security systems are vital additions for smart homes to prevent the financial and physical damages caused by home burglary activities. Most of the currently available vision-based home security systems use face recognition at the entrance of indoor environments. The architecture of such systems is easily forgeable by smart burglars. Not only the architecture but also the methodology employed to implement the system are not robust enough to stand against challenges made by burglars.

This study addressed the highly deceivability problem of the current in-house security systems by designing and implementing a fully vision-based integrated architecture that incorporates both face recognition and human motion detection modules. To achieve the aim of designing and implementing the robust architecture efficient module integration logic with an efficient methodological approach is formulated. In the architecture two modules cooperate together in a parallel fashion making the system capable of providing dual-layer security service. The result is acquired by conducting a vulnerability assessment on both the architecture and the methodological approach of a proposed system using different real-time scenarios. From the scenario-based evaluation result, the proposed architecture reduced the robustness failure of the previous systems with 40%.

As far as the methodological approach is concerned this thesis employed comparatively best algorithms for both modules. The outstanding DNN face detection method employed makes the system to work correctly under different challenges. Besides, the proposed novel combination of the FaceNet system with Logistic Regression classifier for the face recognition subsystem shown a better accuracy of 98.3%.

This is a great performance than other competing methods experimented and the highest accuracy report of the related work method LBP-SVM-PSO [27] as well.

From the literature review, we can understand that most of the related works have used a simple background method to segment a foreground mask of motion image to be classified using HOG and SVM combination. And the highest accuracy achieved is a recognition rate of 87.2%. This study used Adaptive Background Subtraction with the same classification method as previous works but with efficient parameter tuning and achieved 90% accuracy. In general, the overall results of the study show that the problems stated in section *1.3* of the document are solved by developing a security system that is by far reliable than before.

7.2 Future Works

A vision-based indoor environment security system has a long way to go. This study suggests the following paths to be included in the future.

1. **Unusual Activity Analysis:** As already discussed the proposed system provides dual-layer security for the area under surveillance. To make the system even more robust in a way that it can almost avoid false alarms, the natural step after this thesis is to incorporate a Human Activity Recognition (HAR). Even if it is a young area of computer vision, HAR is a very powerful scene analysis technique for surveillance applications. Incorporating HAR also has a great role to customize the system for the application of both restricted environment (where only authorized personnel are allowed to enter) and open environments (where anybody is allowed like shopping malls, banks and etc.)
2. **Computational Time:** In future different aspects of the design and hardware setup will be further studied to achieve better computational performance even on resource-limited devices.
3. **Night Vision:** As already discussed in the scope section, the proposed system concentrated on day time vision excluding the night time vision. Incorporating a night time vision will enable the security system to provide 24-hour monitoring. Therefore, this should be one of the main future directions of this Thesis.

REFERENCES

[1] A. Wilson, J. P. Arun, and A. Jayakrishnan, “Security Alert Using Face Recognition,” *Int. Res. J. Eng. Technol.*, vol. 4, no. 4, pp. 687–691, 2017.

[2] J. S. Kartik, K. R. Kumar, and V. S. Srimadhavan, “SECURITY SYSTEM WITH FACE RECOGNITION , SMS ALERT AND EMBEDDED NETWORK VIDEO MONITORING TERMINAL,” vol. 2, no. 5, pp. 9–19, 2013.

[3] S. S. Shingne and K. V, “Security System Design Based on Human Face Detection and Recognition on Android Platform,” *Ijireeice*, vol. 3, no. 7, pp. 152–157, 2015.

[4] S. N. V, R. Vijayadev, and U. S. Rani, “Real time Face Detection and Recognition in Video Surveillance,” *Int. Res. J. Eng. Technol.*, 2017.

[5] “The Hard Number of Home Invasion,” 2019. [Online]. Available: https://www.alarms.org/burglary-statistics/. [Accessed: 20-Jul-2019].

[6] M. R. Mulla, R. P. Patil, and S. K. Shah, “Facial image based security system using PCA,” *Proc. - IEEE Int. Conf. Inf. Process. ICIP 2015*, pp. 548–553, 2016.

[7] D. Chowdhry, R. Paranjape, and P. Laforge, “Smart Home Automation System for Intrusion Detection,” pp. 75–78, 2015.

[8] D. A. R. Wati and D. Abadianto, “Design of face detection and recognition system for smart home security application,” *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua, pp. 342–347, 2018.

[9] N. D. and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” vol. 2017-Octob, no. Nips, pp. 1–15, 2018.

[10] S. Somani, P. Solunke, S. Oke, P. Medhi, and P. P. Laturkar, “IoT Based Smart Security and Home Automation,” *Asian J. Converg. Technol. (AJCT).*, vol. 5, no. 2350, pp. 1–4, 2019.

[11] K. C. Sahoo and U. C. Pati, “IoT based inSahoo, K. C., & Pati, U. C. (2018). IoT based intrusion detection system using PIR sensor. RTEICT 2017 - 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings, 2018-Janua, 1641,” *RTEICT 2017 - 2nd IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol. Proc.*, vol. 2018-Janua, pp. 1641–1645, 2018.

[12] W. Audette and D. Kynor, “Improved Intruder Detection Using Seismic Sensors and Adaptive Noise Cancellation,” *... Tunn. Detect. ...*, no. Peck 2004, 2009.

[13] N. Surantha and W. R. Wicaksono, “Design of Smart Home Security System using Object Recognition and PIR Sensor,” *Procedia Comput. Sci.*, vol. 135, pp. 465–472, 2018.

[14] G. Yatman, S. Üzumcü, A. Pahsa, and A. A. Mert, “Intrusion detection sensors used by electronic security systems for critical facilities and infrastructures: a review,” *Saf. Secur. Eng. VI*, vol. 1, pp. 131–141, 2015.

[15] S. Tanwar, P. Patel, K. Patel, S. Tyagi, N. Kumar, and M. S. Obaidat, “An advanced Internet of Thing based Security Alert System for Smart Home,” *IEEE CITS 2017 - 2017 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 25–29, 2017.

[16] A. Jain, S. Basantwani, O. Kazi, and Y. Bang, “Smart surveillance monitoring system,” *2017 Int. Conf. Data Manag. Anal. Innov. ICDMAI 2017*, pp. 269–273, 2017.

[17] P. Vigneswari, V. Indhu, R. R. Narmatha, A. Sathinisha, and J. M. Subashini, “Automated Security System using Surveillance,” *Int. J. Curr. Eng. Technol.*, vol. 55, no. 22, pp. 2277–4106, 2015.

[18] H. B. A. R. Mior Mohammad, “Active infrared motion detector for house security system mior mohammad hafiizh bin abd. rani universiti malaysia pahang,” 2013.

[19] S. N. Yoannan, “Security System Based on Ultrasonic Sensor Technology,” *IOSR J. Electron. Commun. Eng.*, vol. 7, no. 6, pp. 27–30, 2013.

[20] T. Kumagai, S. Sato, and T. Nakamura, “Fiber-optic vibration sensor for physical security system,” *Proc. 2012 IEEE Int. Conf. Cond. Monit. Diagnosis, C. 2012*, no. May, pp. 1171–1174, 2012.

[21] D. THAKKAR, “Problems with Current Security Systems That You Should Know,” *BAYOMETRIC*. [Online]. Available: https://www.bayometric.com/problems-current-security-systems/. [Accessed: 20-Jun-2019].

[22] A. P. T. and M. A. Pentland, “Face Recognition using eigenfaces and SVMs,” *Cin.Ufpe.Br*, pp. 586–591, 1991.

[23] N. A. Othman and I. Aydin, “A face recognition method in the Internet of Things for security applications in smart homes and cities,” *Proc. - 2018 6th Int. Istanbul Smart Grids Cities Congr. Fair, ICSG 2018*, pp. 20–24, 2018.

[24] G. Ahmed and B. Mohamed, “Improving the Recognition of Faces using PCA and SVM Optimized by DWT,” *Int. J. Comput. Appl.*, vol. 107, no. 17, pp. 7–11, 2014.

[25] A. A. A. Al-modwahi, O. Sebetela, L. N. Batleng, B. Parhizkar, and A. H. Lashkari, “Facial Expression Recognition Intelligent Security System for Real Time Surveillance,” *World Congr. Comput. Sci. Comput. Eng. Appl. Comput.*, pp. 1–8, 2012.

[26] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, “Facial-based intrusion detection system with deep learning in embedded devices,” *ACM Int. Conf. Proceeding Ser.*, pp. 64–68, 2018.

[27] A. Upasana, B. Manisha, G. Mohini, and K. Pradnya, “Real Time Security System using Human Motion Detection,” vol. 4, no. 11, pp. 245–250, 2015.

[28] Kazarian Artem et al., “Structure and model of the smart house security system using machine learning methods,” pp. 7–10, 2017.

[29] M. H. Ali, F. Kamaruzaman, M. A. Rahman, and A. A. Shafie, “Automated secure room system,” *2015 4th Int. Conf. Softw. Eng. Comput. Syst. ICSECS 2015 Virtuous Softw. Solut. Big Data*, pp. 73–78, 2015.

[30] A. Beatrice Dorothy, S. Britto Ramesh Kumar, and J. Jerlin Sharmila, “IoT Based Home Security through Digital Image Processing Algorithms,” *Proc. - 2nd World Congr. Comput. Commun. Technol. WCCCT 2017*, pp. 20–23, 2017.

[31] N. P. G. Khushbu H Mehta, “Vision Based – Real Time Monitoring Security System for Smart Home,” *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. Vol. 4, no. Issue 2, pp. 1–7, 2016.

[32] Sameerchand Pudaruth, Faugoo Indiwarsingh, and Nandrakant Bhugun, “A Unified Intrusion Alert System using Motion Detection and Face Recognition,” *2nd Int. Conf. Mach. Learn. Comput. Sci.*, vol. 2, pp. 17–20, 2013.

[33] J. See and S. W. Lee, “An integrated vision-based architecture for home security system,” *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 489–498, 2007.

[34] S. Patidar, A. P. Pandey, K. Ketan, and G. R. Pareshkumar, “Real Time Vision Based Security System,” *IOSR J. Electron. Commun. Eng.*, vol. 9, no. 5, pp. 46–53, 2014.

[35] “IMAGNET,” 2016. [Online]. Available: http://www.image-net.org/. [Accessed: 20-Sep-2008].

[36] “OpenCV,” *2018-11-18*. [Online]. Available: https://opencv.org/about/. [Accessed: 20-

Feb-2019].

[37] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks,” *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 27-Mar-2019].

[38] V. Gupta, “Face Detection – OpenCV, Dlib and Deep Learning (C++ / Python),” *Learn OpenCV*. [Online]. Available: https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/.

[39] J. Yi, X. Mao, L. Chen, Y. Xue, A. Rovetta, and C. D. Caleanu, “Illumination normalization of face image based on illuminant direction estimation and improved Retinex,” *PLoS One*, vol. 10, no. 4, pp. 1–20, 2015.

[40] I. A. Kakadiaris, G. Passalis, G. Toderici, T. Perakis, and T. Theoharis, “Face Recognition, 3D-Based,” *Encycl. Biometrics*, pp. 476–485, 2015.

[41] A. Goyal, A. Bijalwan, and M. Chowdhury, “A comprehensive review of image smoothing techniques,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 1, no. 4, pp. 315–319, 2012.

[42] B. Huang, “Review of FaceNet : A Unified Embedding for Face Recognition and Clustering,” pp. 1–2, 2017.

[43] A. Shahid *et al.*, “Computer Vision Based Intruder Detection Framework (CV-IDF),” pp. 2–6, 2017.

[44] J. Lee and M. Park, “An adaptive background subtraction method based on kernel density estimation,” *Sensors (Switzerland)*, vol. 12, no. 9, pp. 12279–12300, 2012.

[45] D. Fradkin and I. Muchnik, “Support Vector Machines for Classi cation,” vol. 0000, pp. 1–9, 2000.

[46] P. Banerjee and S. Sengupta, “Human motion detection and tracking for video surveillance,” *Proc. Natl. Conf. Track. video Surveill. Act. Anal.*, pp. 88–92, 2018.

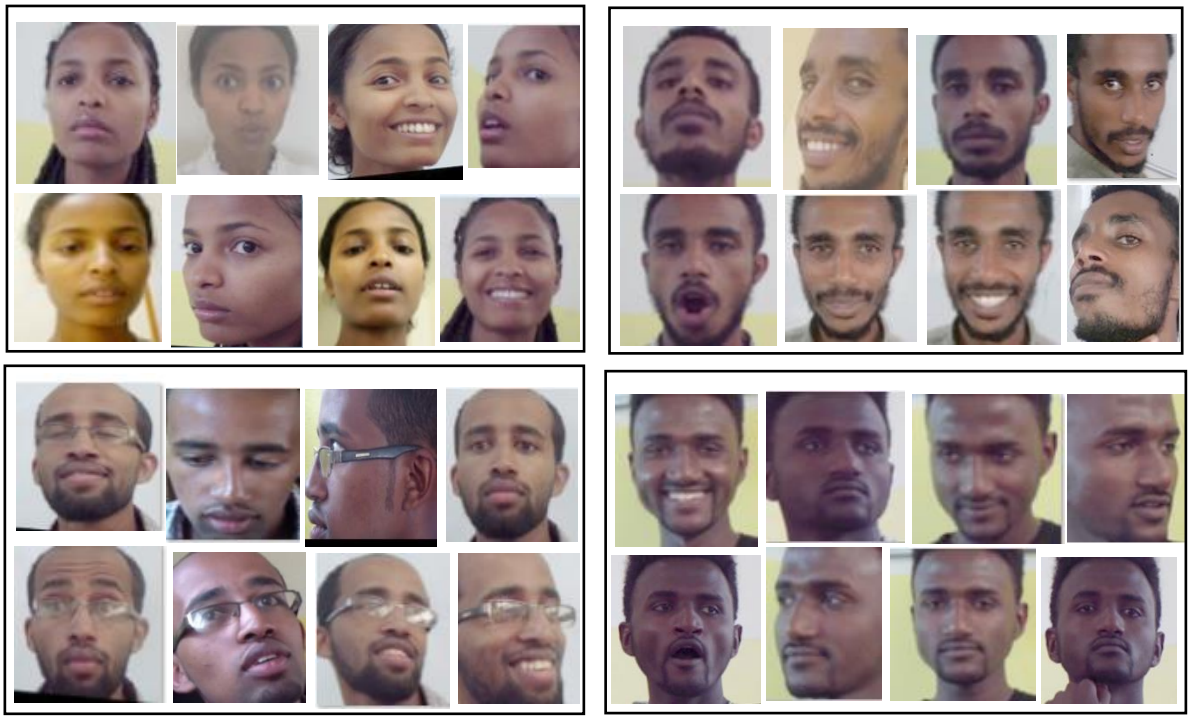
[47] S. S. Bhakar and T. S. Sikarwar, “Guide to Write Research Paper,” no. March, 2015.

[48] A. Shahrokni, *Thesis for The Degree of Doctor of Philosophy Software Robustness: From Requirements to Verification*. 2013.

APPENDIXES

APPENDIX A: Sample Dataset

A1: Sample Face Dataset



A2: Sample Dataset for Human Motion Detection

APPENDIX B: Sample Python Codes

B1: Python Code for Face Detection

```
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"],
args["model"])
print("[INFO] reading a video frame...")
if video_file:
    webcam = cv2.VideoCapture(args["video"])
else:
    webcam = cv2.VideoCapture(0)
if(not webcam.isOpened()):
    print("Error opening webcam")
    exit()
while(webcam.isOpened()):
        status, frame = webcam.read()
    if(not status):
        print("Error reading frame")
        exit()
    print("[INFO] resizing a frame...")
    frame = imutils.resize(frame, width=500)
    # grab the frame dimensions and convert it to a blob
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300,
           300)), 1.0, (300, 300), (104.0, 177.0, 123.0))
    net.setInput(blob)
    detections = net.forward()
    # loop over the detections
    id = 0
    for i in range(0, detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        # filter out weak detections
        if confidence < args["confidence"]:
            continue
```

```
            box = detections[0, 0, i, 3:7] * np.array([w, h, w,
                  h])
            (startX, startY, endX, endY) = box.astype("int")
            # initialized the detection probability
            text = "{:.2f}%".format(confidence * 100)
            y = startY - 10 if startY - 10 > 10 else startY + 10
            # crop the ROI to be saved
            crop=frame[startY:endY,startX:endX]
            crop=cv2.resize(crop, (256,256))
            #draw rectangle along with detection probability
            cv2.rectangle(frame, (startX, startY), (endX,
            endY),(0, 0, 255), 2)
            cv2.putText(frame, text, (startX, y),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
            #write on adisk
            path="Detected"
            count=0
            for file_type in [path]:
                for img in os.listdir(file_type):
                    count+=1
            id = count + 1
            imgName="tr"+str(id)
            crop_width, crop_heigh, channal = crop.shape
            print('Width of corp', crop_width)
            print('Height of crop', crop_heigh)
            if crop_width > 0 and crop_heigh > 0:
                cv2.imwrite("./Detected/"+imgName+".jpg",crop)
        # show the output frame
        cv2.imshow("Frame", frame)
        if (cv2.waitKey(1) & 0xFF) == ord('q'):
            break
cv2.destroyAllWindows()
webcam.release()
```

B2: Python Code for Face Recognition from Test Video

```
print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"],
args["model"])
embedder = cv2.dnn.readNetFromTorch(args["embedding_model"])
recognizer = pickle.loads(open(args["recognizerF"],
"rb").read())
le = pickle.loads(open(args["leF"], "rb").read())
predictor = dlib.shape_predictor(args["shape_predictor"])
fa = FaceAligner(predictor, desiredFaceWidth=256)
vs = VideoStream(src=1).start()
time.sleep(2.0)
while True:
    frame = vs.read()
    if frame is None:
        break
    image = imutils.resize(frame, width=500)
    (h, w) = image.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(image, (300,
    300)), 1.0, (300, 300), (104.0, 177.0, 123.0))
    net.setInput(blob)
    detections = net.forward()
     for i in range(0, detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence < args["confidence"]:
            continue
     box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")
        face = image[startY:endY, startX:endX]
   faceOrig = imutils.resize(image[StartY: StartY + endY,
   startX: startX + endX], width=256)
   faceAligned = fa.align(image, faceOrig, box)
    print("gamma correcting....")
    adjusted = adjust_gamma(faceAligned, gamma=1.5)
```

```
    adjusted=cv2.GaussianBlur(adjusted, (13,13),
    cv2.BORDER_DEFAULT)
    faceBlob = cv2.dnn.blobFromImage(adjusted, 1.0 / 255,(96,
    96), (0, 0, 0), swapRB=True, crop=False)
     embedder.setInput(faceBlob)
     vec = embedder.forward()
     preds = recognizer.predict_proba(vec)[0]
        j = np.argmax(preds)
        proba = preds[j]
        if proba<=0.20:
            name=="Unknown"
        else:
            name = le.classes_[j]
        text = "{}: {:.2f}%".format(name, proba * 100)
        startY = startY - 10 if startY - 10 > 10 else startY
        + 10
        cv2.rectangle(image, (startX, startY), (endX,
        endY),(0, 0, 255), 2)
        cv2.putText(image, text, (startX, startY),
        cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
    # show the output frame
    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
vs.stop
cv2.destroyAllWindows()
```

B3: Python Code for Motion Segmentation

```
subtractor = cv2.createBackgroundSubtractorMOG2
(history=60,VarThreshold=50,detectShadow=false)
while True:
    frame = vs.read()
    if frame is None:
        print("No frame")
        break
    # resize the frame and apply Background Subtractor
    frame = imutils.resize(frame, width=500)
    orig=frame.copy()
    fgmask = subtractor.apply(frame)
    thresh = cv2.threshold(fgmask, 50, 255,
    cv2.THRESH_BINARY)[1]
    kernel=np.ones((5,5), np.uint8)
    erosion=cv2.erode(thresh, kernel, iterations=1)
    dialation = cv2.dilate(erosion, kernel, iterations=1)
    opening=cv2.morphologyEx(dialation, cv2.MORPH_OPEN, None)
    closing=cv2.morphologyEx(opening, cv2.MORPH_CLOSE, None)
    cnts1 = cv2.findContours(closing.copy(),
    cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    cnts1 = imutils.grab_contours(cnts1)
    for c in cnts1:
        if cv2.contourArea(c) < args["min_area"]:
            continue
        (x, y, w, h) = cv2.boundingRect(c)
        Crop=frame[y:y + h, x:x + w]
        Crop=cv2.resize(Crop,(64, 128))
        y = y - 10 if y - 10 > 10 else y + 10
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255,
        0), 2)
```

B4: Python Code for Motion Classification from Test Video

```
gray = cv2.cvtColor(Crop, cv2.COLOR_BGR2GRAY)
edged = imutils.auto_canny(gray)
cnts2 = cv2.findContours(edged.copy(),
cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts2 = imutils.grab_contours(cnts2)
if cnts2:
    c = max(cnts2, key=cv2.contourArea)
else:
    continue
(x1, y1, w1, h1) = cv2.boundingRect(c)
entity = gray[y1:y1 + h1, x1:x1 + w1]
entity = cv2.resize(entity, (200, 100))

# Histogram of Oriented Gradients descriptor from the
ROI
H = feature.hog(entity, orientations=9,
pixels_per_cell=(4, 4),
    cells_per_block=(2, 2), transform_sqrt=True,
    block_norm="L1")
width = int(Crop.shape[1] * 2)
height = int(Crop.shape[0] * 2)
dim = (width, height)

# load the motion recognizer and label encoder
model = pickle.loads(open(args["recognizer"],
"rb").read())
le = pickle.loads(open(args["le"], "rb").read())
```

```
pred=model.predict_proba(H.reshape(1,-1))[0]
j = np.argmax(pred)
proba = pred[j]
name = le.classes_[j]
text = "{}: {:.2f}%".format(name, proba * 100)
cv2.putText(frame, text, (x-10, y-5),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
cv2.putText(frame, text, (x, y+100),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

cv2.putText(frame,datetime.datetime.now().strftime("%
A %d %B %Y %I:%M:%S%p"),(10, frame.shape[0] - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0, 255), 1)
```