



CCTV Intruder Detection System

**A dissertation submitted for the Degree of Master of
Information Technology**

G.H.A.Chaminda

University of Colombo School of Computing

2019



Abstract

With the increasing complexity associated with modern life, people's concern for safety has assumed top priority in their daily list of needs. In order to ensure their safety and to lead a peaceful life, they have developed and applied a variety of security systems, which have different features and equipment. Normally, there are four major security systems used in important places. They include closed circuit television (CCTV), alarm system, access control system and fire alarm system. Thus, choosing the appropriate system depends on the desired purpose. Among these alternatives, use of CCTV is a popular option adopted by many people in order to ensure the safety of their home front and this has currently been developed from Analog system to IP system. However, at present, the system still lacks the ability to alert users timely and this drawback results in property damages as well as risks to human life.

Therefore, in this project, it is proposed to develop an indexing system to find relevant events in a CCTV recording. The whole project consists of four sub models such as Image acquisition model, Image processing model, Information extraction model and E-mail generation stage. The proposed project on CCTV Intruder Detection system can successfully identify the intruders, minimize the false alarms and minimize the human supervision of such systems. It automatically detects the unauthorized activities on the premises: for example, it detects unauthorized access to the premises and sends the warning message to the system owner. Then the system owner can respond according to the incident by remotely logging on to the system and viewing CCTV live stream, playing an alarm, calling the police, etc. This system is highly recommended for the home surveillance which has become an integral part of home security system. The technologies used in this project are OpenCV 3.4.1 and Python 3.0.

The project has a distinctive edge in terms of time and efficiency when compared with similar systems discussed here. That is, in an ordinary surveillance system, in case of an incident, all the recorded videos should be manually viewed in order to find the intrusion. This consumes lot of time. In contrast, in the present system, an immediate detection can be obtained because of the email that is generated for the user giving such information as time and image of intrusion. Further, the system has another advantage as it is developed to minimize false alarms. This is achieved through adjusting the threshold pixel value of the image of the intruder. The user can also watch the intrusion through a live video as a follow up and this is another outcome of the project.

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: G.H.A.Chaminda

Registration Number: 2016/MIT/009

Index Number: 16550094

Signature:

Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Signature:

Date:

Acknowledgements

I would first like to thank my project supervisor Dr. Ajantha Atukorale, Deputy Director of the School of Computing at University of Colombo. The door to Dr. Atukorale's office was always open whenever I ran into a trouble spot or had a question about my project implementation. He constantly helped and advised me to complete this project successfully.

I would also like to thank the other experts who helped me in the image processing part of this project: Mr. Prabath Samarasinghe (Lecturer of University of Sri Jayewardenepura), Mr. Roshan Rajapakse (Lecturer of University of Colombo). Without their help, the implementation of the system could not have been successfully conducted.

I would also like to acknowledge Dr. Lalith Ananda, Senior Lecturer, Department of English, University of Sri Jayewardenepura for English editing of this thesis, and I am gratefully indebted to him for his service.

Finally, I must express my very profound gratitude to my mother, my wife and classmates for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of project implementation and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

	Page No
1.0 Chapter 1 - Introduction	
1.1 Chapter Overview	1
1.2 Background	1
1.3 Motivation	2
1.4 Aim	2
1.5 Objectives	3
1.6 Scope of the Project	3
1.7 Features of the CCTV Intruder detection system	4
1.8 Information sources for the project	4
1.9 Limitations	4
1.10 Structure of the dissertation	5
2.0 Chapter 2 – Background	
2.1 Chapter Overview	6
2.2 Previous work	6
2.3 Technical comparison of proposed system with previous projects	12
2.4 Chapter Summary	13
3.0 Chapter 3 – Methodology	
3.1 Chapter Overview	14
3.2 Image processing in Computer Vision	14
3.3 The hardware components setup	15
3.4 Design Preparation	16
3.5 Design Methodology	16
3.6 Methodology Justification	17
3.7 Overall Design of the system	17
3.8 The high level architecture of the system	18
3.9 Internal functions of the system	22
3.10 Coding for the important steps used in OpenCV	22
3.11 Chapter Summary	23

4.0 Chapter 4 – Testing and Evaluation	
4.1 Chapter Overview	24
4.2 Testing methods	24
4.3 Description of the testing procedure	25
4.4 Test Cases	25
4.5 System performance evaluation	27
4.6 Chapter summary	30
5.0 Chapter 5 – Conclusion	
5.1 Introduction	31
5.2 The evaluation of the results of the project	31
5.3 Things achieved via this project	32
5.4 Problems encountered in the project	32
5.5 Outcomes of the Project	33
5.6 Future work	33
References	34
Appendix A : Python Source Code	35
Appendix B : User Manual	41

List of Figures

Page No

Figure 2.1 : A.Chiang, J.Chu and S.Su - proposed system of the Intruder Detection ...	7
Figure 2.2.: Edge detection	11
Figure 2.3 : Find real Edges	11
Figure 2.4 : Technical comparison	12
Figure 3.1: Hardware setup	15
Figure 3.2 : Main Stages of the System	17
Figure 3.3 : System Architecture	18
Figure 3.4 : Use case Diagram	19
Figure 3.5 : Class Diagram	20
Figure 3.6 : Sequence Diagram	21
Figure 4.1 : Original Frame Before Background Subtraction	28
Figure 4.2 : Results of Background SubtractorMOG	28
Figure 4.3 : Results of Background SubtractorMOG2	29
Figure 4.4 : Results of Background SubtractorGMG	29
Figure 7.1 : Command Prompt	41
Figure 7.2 : User Interface Of the Project	42
Figure 7.3 : Intruder Detection 1	43
Figure 7.4 : Intruder Detection 2	43
Figure 7.5 : Received Email	44
Figure 7.6 : Attachment of the Received Email	44

List of Abbreviations

2-D	-	Two Dimensional
3-D	-	Three Dimensional
AI	-	Artificial Intelligence
CCTV	-	Closed Circuit Television
CPU	-	Central Processing Unit
FPS	-	Frames per Second
GUI	-	Graphical User Interface
HOQ	-	Histogram of Oriented Gradients
HSV	-	Hue Saturation Value
IEEE	-	Institute of Electrical and Electronics Engineers
IMAQ	-	Image Acquisition
MIT	-	Master of Information Technology
ML	-	Machine Learning
MPI	-	Message Passing Interface
NI Vision	-	National Instruments Vision
OpenCV	-	Open Computer Vision
OpenMP	-	Open Multi Processing
RAD	-	Rapid Application Development
RGB	-	Red Green Blue
RUP	-	Rational Unified Process
SSADM	-	Structured System Analysis and Design Method
UCSC	-	University of Colombo School of Computing
VGA	-	Video Graphics Accelerator
VI	-	Virtual Interface
VM	-	Virtual Machine

Chapter 1

Introduction

1.1 Chapter Overview

This chapter presents the background to the study and the author's inspiration and motivation behind the project. Project background, aims and objectives are explained so as to give a brief insight regarding the requirements in initiating this project.

1.2 Background

Closed circuit Television (CCTV) systems are the most popular and widely used technology for intruder detection or surveillance worldwide. These systems normally give live video for the terminal and 24-hour video recording. Therefore, in most situations someone needs to observe the live video. If not, when an incident happens, people need to watch all the videos of the past known time period to get an idea of the particular incident. Hence, this system is not an efficient one for the present world. Therefore, this project mainly focused on identifying intruders and at the same time sending the alert message to the system owner. According to the alert generated time it is easy to identify the correct time frame of the incident and further, system owner can watch live video immediately after the incident. Then system owner can decide to play an alarm or call the police.

This system is highly recommended for the home surveillance. Home surveillance has become an integral part of our home security system. There are numerous occasions when one has to leave home for several days and engage in work in his or her work place or elsewhere. In such circumstances taking care of your loved ones and property become a matter of outmost importance. These are the main reasons for the increasing demand for such systems.

Most of the currently used burglar alarm systems and CCTV Intruder detection systems use some sensors such as Motion detection sensors, Passive infrared detectors, Ultrasonic detectors, Microwave detectors, Photo electric beams, Glass break detection sensors, Smoke heat and Carbon Monoxide detectors, etc. to detect the intruders. Some systems use hybrid systems (Several sensors) for further enhancement of accuracy.

However, image processing principle is less commonly applied in market products. This could be due to a number of reasons such as its sole dependence on software system and its relative complexity as a method when compared with sensing technologies.

1.3 Motivation

This project on CCTV Intruder Detection system was inspired as a result of this researcher's interest in Image processing, Face detection and recognition and Artificial Intelligence (AI). A further motivation was the general conviction that image processing and artificial intelligence would be the future of novel research projects and technology and thereby would contribute significantly to the development of state of the art mobile phones, security systems, sophisticated weaponry etc. The advanced mathematical formulas that underlie all these subject areas were also motivating factors for this author as he has a special interest to further study the correlation between those and CCTV Intruder Detection system, Image processing, Face detection and recognition and Artificial Intelligence.

1.4 Aim

This project on CCTV Intruder Detection system is proposed to successfully identify the intruders, minimize the false alarms, and minimize the human supervision of such systems. It automatically detects the unauthorized activities on the premises: for example, it detects unauthorized access to the premises and sends the warning message to the system owner. Then the system owner can respond according to the incident by remotely logging on to the system and viewing CCTV live stream, playing an alarm, calling the police, etc.

The overall system has been developed with following features.

- ✚ Low Cost – Select low cost devices (Camera and others), open source software and internet service provider for implementation. In addition, it was also planned to reduce implementation and maintenance cost of the product.
- ✚ Reliability – It should provide robust service under any circumstances.
- ✚ User friendliness – Easy to operate and learn the system. (Simple GUI is used.)

1.5 Objectives

The project mainly focuses upon the following objectives:

- To identify the basic image processing principles and theories.
- Refer the Open CV software's Image processing tools and identify their characteristics. This was followed by further study on how to detect differences on previous image with new images.
- To automate the system to detect the variations in scene and react with automatic reply according to the given scenarios.
- To integrate the system in Windows / Linux based platform. This was followed by remotely accessing the system and doing necessary activities remotely.
- To provide an indexing system to find relevant events in a CCTV recording.

1.6 Scope of the Project

The scope of the project mainly extends to such areas as image capturing, image processing, and image comparison. A number of other steps and stages are also included in these main tasks. These are image storage, transmission: digitization/quantization, compression, and encoding/decoding. The project also covers image enhancement and restoration for improvement of pictorial information. Further, it will include information extraction for further computer analysis. According to the results, email generation will also cover the scope of the project.

1.7 Features of the CCTV Intruder Detection System

- ✓ The main purpose of this project is to identify the incident and give immediate response. Therefore, image processing principals are used to identify differences between images. (It always checks the newly fed video with stored image.)
- ✓ Windows / Linux based application developed to remotely connect to the system.
- ✓ Send automated E-mail to particular users.

1.8 Information sources for the project

- Contribution from the Supervisor and Resource persons.
- The University of Colombo School of Computing (UCSC) Library
- Previous projects, Research Articles and Newspapers
- Internet

1.9 Limitations

This project was inspired by the author's interest in image processing and artificial intelligence. It was also inspired by the author's understanding that Image processing and artificial intelligence would be the future of novel research projects and technology. Unfortunately these subject areas are not included in the Master of Information Technology (MIT) syllabus. These subject areas are still novel and still being explored and researched. Therefore, dearth of sufficient literature on these subject areas was a limitation. However, this author could gain a sufficient knowledge and information on these areas by reading text books, previous articles and projects. A further limitation was lack of computers with sufficient processing power to process continuous streaming of images. Also, the existence of CCTV Camera systems which are not compatible with the Windows and Linux operating systems was another limitation.

1.10. Structure of the Dissertation

Chapter 1 is mainly focused on presenting a brief introduction to the project and the background to initiate this project. It further discusses the aims and objectives, features and information sources referred to doing the project.

Chapter 2 presents the background to the study. This also includes a brief overview of previous research projects which have relevance to the proposed one. This will be followed by a technical comparison of proposed system with previous projects.

Chapter 3 focuses on discussing the methodology adopted in the project. It describes how image processing is done by using computer vision, hardware components used for the project, design methodology and justification of why such methodology was used, high level architecture of the system, internal functions of the system and coding samples used for image processing.

Chapter 4 is based on Testing and evaluation of the system. It presents the testing methods, testing procedure, test cases and system performance evaluation tasks.

Final chapter is dedicated to conclusion and it provides a clear insight into things achieved via this project, problems encountered and how to tackle the problems and future work needed to improve the project further.

Chapter 2

Background

2.1 Chapter Overview

This chapter presents an overview of existing literature related to the previous projects done by various people by using the image processing techniques for intruder detection. The research studies that have been carried out in this regard are referred to with a view to presenting a holistic picture as well as showing the research gap that exists.

2.2 Previous Work

There are many projects which have been done regarding intruder detection, Image processing, remote sensing and home automation. However, Image processing has not been commercially developed in CCTV intruder detection systems. Most of the systems use several types of sensing techniques like Motion detection sensors, Passive infrared detectors, Ultrasonic detectors, Microwave detectors, Photo electric beams, Glass break detection sensors, Smoke heat and Carbon Monoxide detectors, etc.

2.2.1 Intruder detection system by A.Chiang, J.Chu and S.Su

A project called, “Intruder detection system” which has been supervised by School of Electronics and Applied Sciences of Washington University in St. Louis by A.Chiang, J.Chu and S.Su [1] is quite notable in this regard. They used LabVIEW with NI Vision to build an intruder detection system algorithm to detect the presence of an intruder. Their main concern was to minimize the false alarms that happen due to the pets that come into the viewing space. They used RGB camera to capture images as 25 fps sequence and analyzed through frame comparisons converted to binary images that represent change, and then filtered through a series of image refining VI’s helping to enhance their change detection results.

The basic process of this project is that the camera captures a real time video stream, then the video is fed into Lab View for processing.

The difference between the current frame of the video and its previous frame is analyzed. If there is an intruder in the video, an alert email and text message will be sent by Lab View.



Figure 2.1 : A.Chiang, J.Chu and S.Su - proposed system of the Intruder Detection [1 , fig. 1]

They have done the Image processing part in several steps.

- Subtraction (The current frame of the video is subtracted by its previous frame using the IMAQ Subtract VI.)
- Threshold Image. (Turn the subtracted image into a binary image by using the IMAQ Threshold VI so that each pixel will have a value of either 1 or 0.)
- Filter out noises and fill holes. (Apply the IMAQ RemoveParticle VI and the IMAQ Morphology VI to the binary image to filter out noises, which are small particles. Then apply the IMAQ Convex Hull VI to fill the holes.)
- Clamp the filtered image. (Apply horizontal clamp and vertical clamp to the filtered image to get the height and width of the figure in the image.)
- Draw tracking box. (Draw a box around the moving object based on its height and width. Then overlay the tracking box on the original image for visual appeal to users.)

- The final step is alert. (The height given by the horizontal clamp is first compared with a range of threshold set by designers, which checks if the moving object is a pet. If the height is high enough to fall out of the range, the height is then divided by the width given by the vertical clamp to get the height-width ratio.

The ratio is then compared with another range of threshold set by designers, which is the possible range of human height-width ratio. If the calculated ratio falls into the range, the alarm will be triggered and alerting emails and text messages will be sent to the users.)

2.2.2 Border Security: Intruder Human Detection using Image processing by V.Panchal, H.Dimble, S.Nikam and S.Kamatji

“Border Security: Intruder Human Detection Using Image Processing” conducted by Department of Computer Engineering, University of KJCOEMR, Pune, India by V.Panchal, H.Dimble, S.Nikam and S.Kamatji [2] is a really interesting project related to defense industry. They implemented a new defense mechanism which has a detecting and an aiming unit. Their main objective was to help the armed forces to catch intruders. The targeting unit has a microcontroller along with sensors to detect the presence of humans in the targeted area and process various atmospheric parameters electronically. Laser pointer is used to determine if the desired target is hit or missed by tracking the path of the laser and determining the point of intersection with the laser. Once the parameters are obtained and are used to the targeting unit, it automatically positions the gun using its special mechanism.

Their proposed system can be divided into 3 major parts.

- Object detection. (For detecting objects they have used the Histogram of Oriented Gradients (HOG) descriptor and this helps to train highly accurate object classifiers. Then the descriptor is used to detect human objects in the video stream.

- Motion Tracking. (Once the objects within the image have been detected, the next logical step is to track those objects. They used the algorithm which is known as the Lucas-Kanade algorithm for tracking by optical flow. In this method, they have given some points to track and receive the optical flow vectors of those points.
- Colour detection. (The system will be able to differentiate between allied targets and enemy targets by detecting colours. They used HSV instead of RGB to detect colours.

2.2.3 A Simple intruder detection using a web camera and an image processing by N.Tamaraw and A.Yoshiname

N.Tamaraw and A.Yoshiname [3] developed a system called 'A Simple intruder detection using a web camera and an image processing'. The intruder is detected by comparing the test image with the template that is standard. Both images are divided into small blocks. The authors clarified the relation between the slice level for the determination of the different blocks, the different block number and they divided block number. In order to get the guide, they determined the ideal slice level and the effective block number. Moreover, they considered the computation method with less execution time while the system performance does not deteriorate.

2.2.4 Automated intruder detection system from image sequences using minimum volume sets by T.Ahmed et al.

T.Ahmed et al. [4] Department of Computer science, International Islamic University Malaysia developed automated intruder detection system from image sequences using minimum volume sets. They proposed a new algorithm based on machine learning techniques for automatic intruder detection in visual surveillance networks. The proposed algorithm is theoretically founded on the concept of Minimum Volume Sets. Through application to image sequences from two different scenarios and comparison with existing algorithms, they showed that it is possible for their proposed algorithm to easily obtain high detection accuracy with low false alarm rates.

2.2.5 Edge Detection System by M.Naveenkumar and A.Vadivail

A project called, “Edge Detection System” which has been carried out by Department of Computer Applications of National Institute of Technology, Trichy, Tamilnadu by M.Naveenkumar and A.Vadivail [5] is really helpful to detect edges on objects. In this system they used popular edge detection algorithm called “Canny Edge Detection”. The structural information can be extracted from the input image by applying this technique. It is a multi-stage algorithm. The steps are,

- Noise Reduction (The edge detection is susceptible to noise in the image. Therefore, they used 5x5 Gaussian filter to remove the noise.
- Finding Intensity gradient of the image. (Smoothened image is then filtered with a sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (G_x) and vertical direction (G_y). From these two images, it was aimed to find edge gradient and direction for each pixel.
- Non-maximal suppression (After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. At every pixel, it is checked if it is a local maximum in its neighborhood in the direction of gradient.)

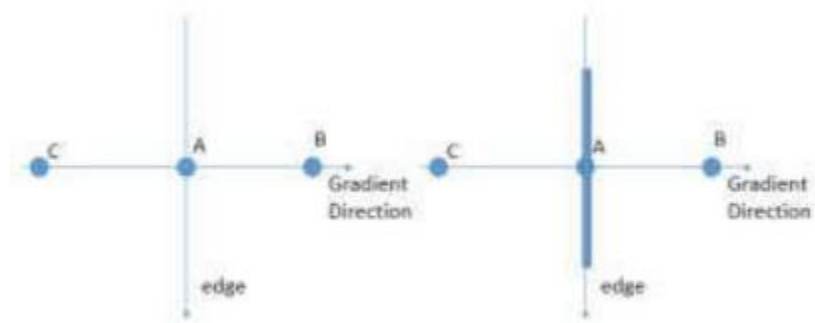


Figure 2.2.: Edge detection [5 , fig. 3]

- Hysteresis Thresholding (At this stage, decides which are all edges are real edges or not.)



Figure 2.3 : Find real Edges [5 , fig. 4]

2.3 Technical comparison of proposed system with previous projects

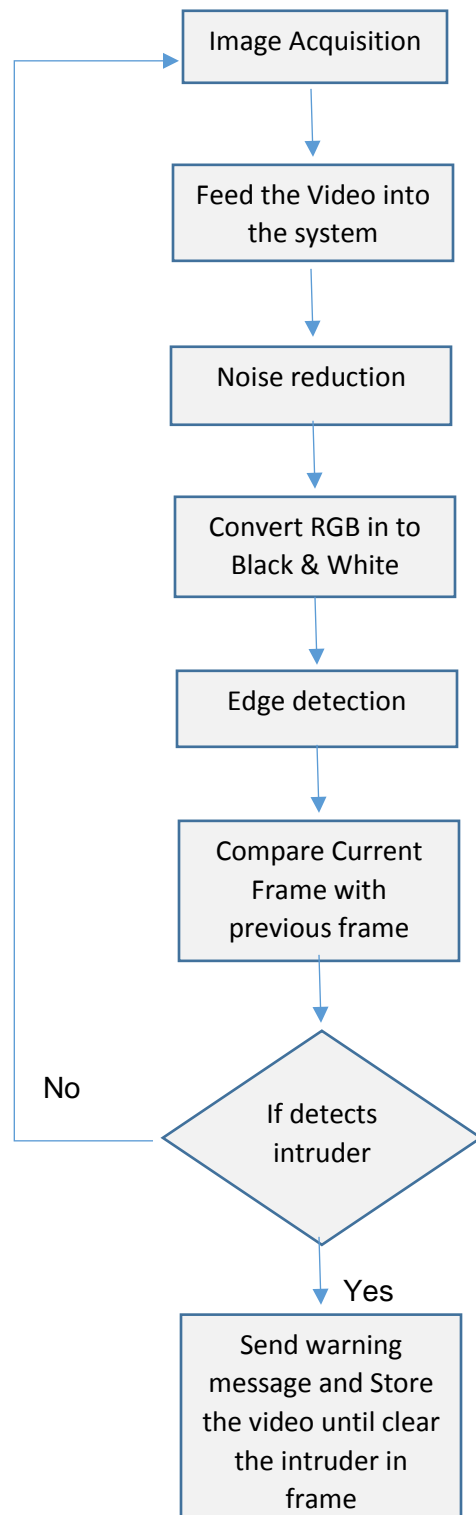


Figure 2.4 : Technical comparison

Most of the previous systems have used above methodology to identify the intruder. All of them have compared currently feeding video (Frame wise) with previously stored image. If some new object appears on the frame, system can easily identify it when subtracting current frame's pixel values with already stored frame. Some of the previous projects have been further improved reducing false alarms due to pets and small animals that appeared in the frame. Up to this point, proposed system has used the same mechanism and OpenCV is used to image processing.

This system further gives live access to the camera and users can easily find the time frame when alarm has been made.

2.4 Chapter Summary

This chapter presented an overview of existing literature related to the previous projects done by researchers by using the image processing techniques for intruder detection. The discussion included the projects carried out by (1) A.Chiang, J.Chu and S.Su, (2) V.Panchal, H.Dimble, S.Nikam and S.Kamatji, (3) N.Tamaraw and A.Yoshiname, (4) T.Ahmed et al. and (5) M.Naveenkumar and A.Vadivail. Although there were slight differences in the methodologies used by them, overall, all such projects focused on image processing for various purposes. A technical comparison of proposed system with previous projects was also presented in this chapter.

Chapter 3

Methodology

3.1 Chapter Overview

This chapter presents the design and the methodological approach taken to achieve the intruder detection. Further it will describe the OpenCV libraries and algorithms used and problems faced due to practical reasons in image capturing and image processing.

The technologies which have been used in this project are OpenCV 3.41 and Python 3.0. OpenCV plays a vital role in the project and Python 3.0 is used as a programming language.

3.2 Image Processing in Computer Vision

OpenCV was started at Intel in 1999 by Gary Bradsky [6] and the first release came out in year 2000. OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day by day. Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc. and is available on different platforms including Windows, Linux, OS X, Android, iOS etc.

Basically there are four modules introduced in OpenCV.

- cv – Main OpenCV functions, image processing and vision algorithms.
- cvaux – Auxiliary (experimental) OpenCV functions.
- cxcore – Data structures, linear algebra support, XML support drawing functions and other algorithms.
- highgui - GUI functions, Image and video I/O.

OpenCV comes with several important libraries and Numpy is a highly optimized library for numerical operations. It gives a MATLAB style syntax. All the OpenCV array structures are converted to-and-from Numpy arrays. Besides that, several other libraries like SciPy, Matplotlib which support Numpy can be used with image processing.

Image processing is a method to perform some operations on an image in order to extract some useful information from it. An image is nothing more than a two dimensional matrix (3-D in case of colored images) which is defined by the mathematical function $f(x,y)$ where x and y are the two co-ordinates horizontally and vertically. The value of $f(x,y)$ at any point gives the pixel value at that point of an image. The pixel value describes how bright that pixel is, and/or what color it should be. For grayscale images the pixel value is a single number that represents the brightness of that pixel. The most common pixel format is the byte image, which is stored as an 8-bit integer giving a range of possible values from 0 to 255. As a convention is taken to be black, and 255 is taken to be white the values in between make up the different shades of gray.

Main tasks in image processing could be categorized in three stages.

- Image acquisition, storage, transmission: digitization/quantization, compression, encoding/decoding.
- Image Enhancement and Restoration: for improvement of pictorial information.
- Information Extraction: for further computer analysis.

3.3 The hardware components setup

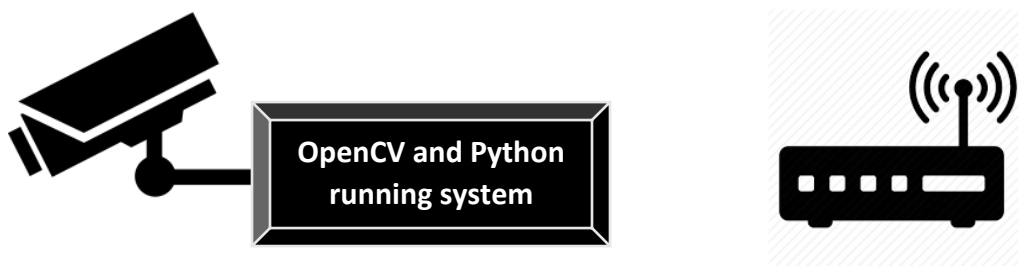


Figure 3.1: Hardware setup

Figure 3.1 represents the hardware architecture of the project. VGA camera is connected to the computer which has separate virtual computer running on Ubuntu operating system. It connects to the Wi-Fi network. This wireless connection is used to connect to the computer in order to get video output. Further it will be used to send E-mails to configured recipients.

3.4 Design Preparation

In devising this template, the author has extracted information from many sources including various texts on software engineering, Object Oriented software development, IEEE documentation standards and online resources. Designing tasks not solely depended on particular standard and to place more emphasis on content than on format.

3.5 Design Methodology

The Rational Unified Process (RUP) based spiral design philosophy is followed backed by Rapid Application Development (RAD) semantics. Spiral methodologies allow prototyping various portions of the design and adjust the design as wanted. Once the design, code and test portions of the final product are completed changes must be evaluated and the same sequence recreated for the new design. This repeats continuously until the final product is completed.

The iterative development cycle is a spiral that passes repeatedly through four phases.

- Plan – The phase is dedicated to the identification of requirements in specific detail as well as the scheme utilized in achieving those requirements.
- Implement – During implementation, a proto type is built to test the solutions arrived at during the planning phase.
- Measure – The prototype is subjected to observation and evaluation of its applicability. It is important to understand that measurements must be both objective and subjective.
- Learn – This is the analysis phase, where one decides which parts of the prototype are doing well and which are not.

There can be many reasons if a prototype does not achieve its goals. Sometimes these will be implementation issues like correct libraries not been used and processing takes more memory and other resources, insufficient understanding of the users, etc. As one makes the transition back to planning this fresh look from a real user's perspective, derived from user testing in the measurement phase is essential.

3.6 Methodology Justification

The Waterfall method (SSADM) and similar conventional techniques are not convenient for the system requirements. As the Image processing and Artificial Intelligence subjects are quite novel and the author too not having sufficient experience on the same, a prototyping based approach would be an ideal solution for the design. The spiral method allowed the developer to approach a problem and solve some sort of it and then make adjustments to suit the design until the output of the project is satisfactory.

Further, a Rapid Application Development methodology suits the project time constraints as well. Due to the limited time frame of the implementation, a systematic and exhaustive approach was not the best selection.

3.7 Overall Design of the System

This section presents a clear idea of the overall design of the project. The whole project consists of four sub models as shown below. Image acquisition model is responsible for providing sequence of images (live video stream) to image processing model. Then the Image processing model grabs all the video frames and do some enhancement of image quality to remove unwanted noise additions. Further it converts the color scheme to Gray mode. Information extraction stage is the next model and its major duty is to find the new objects comparing each frame. Finally E-mail generation stage plays its required role according to the previous stage results.

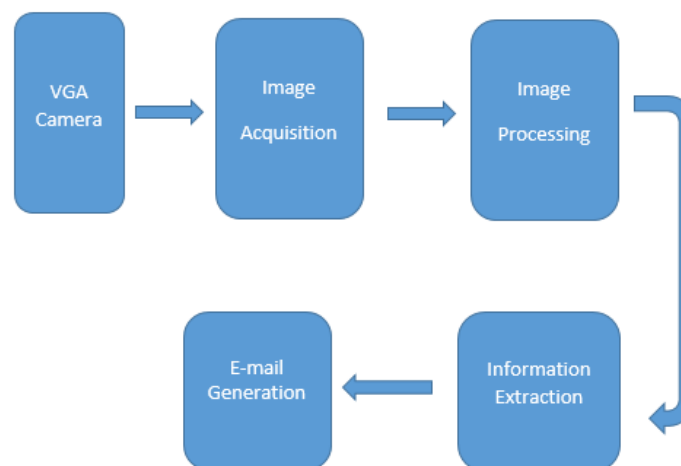


Figure 3.2 : Main Stages of the System

3.8 The high-level architecture of the system

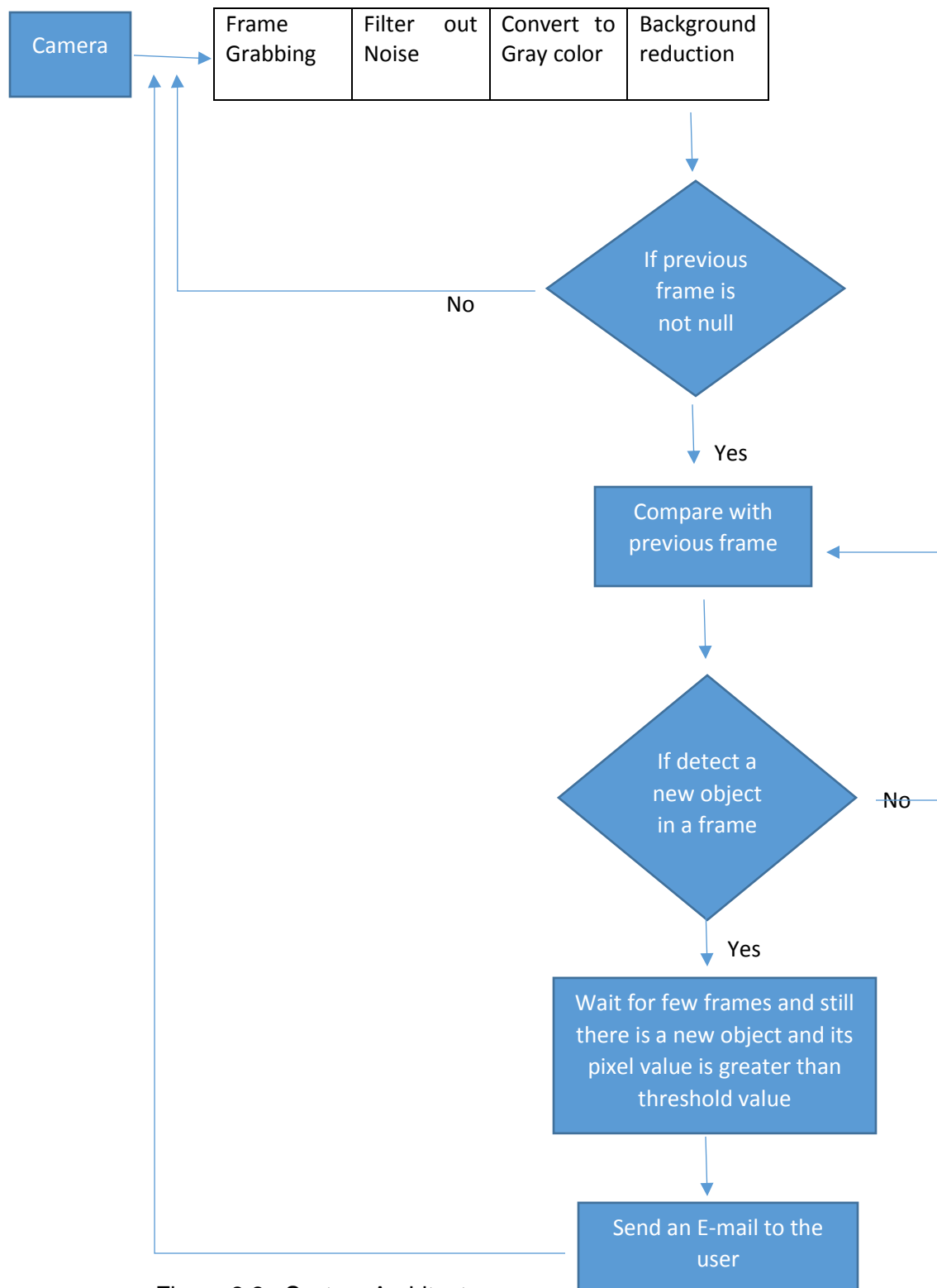


Figure 3.3 : System Architecture

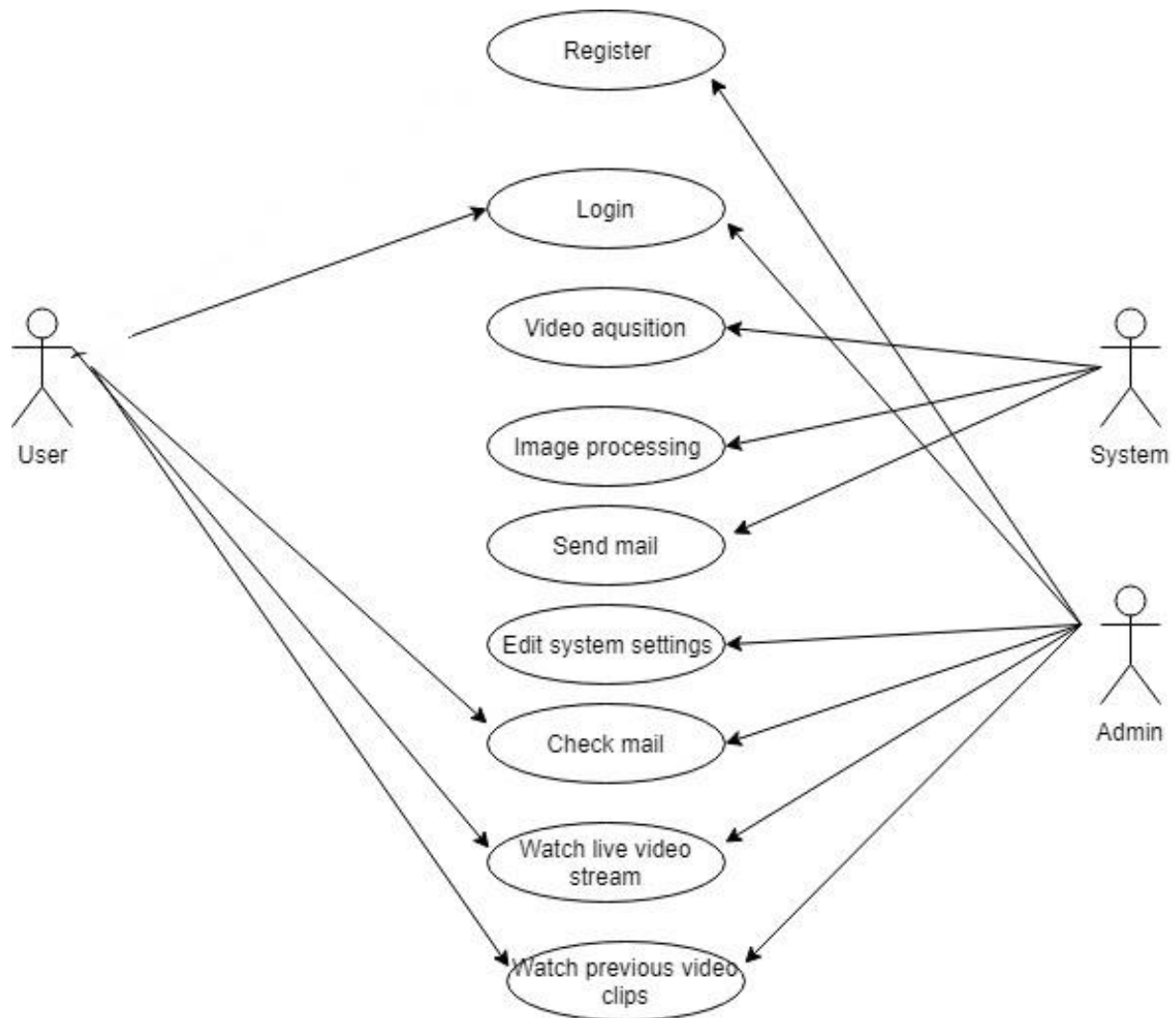


Figure 3.4 : Use case Diagram

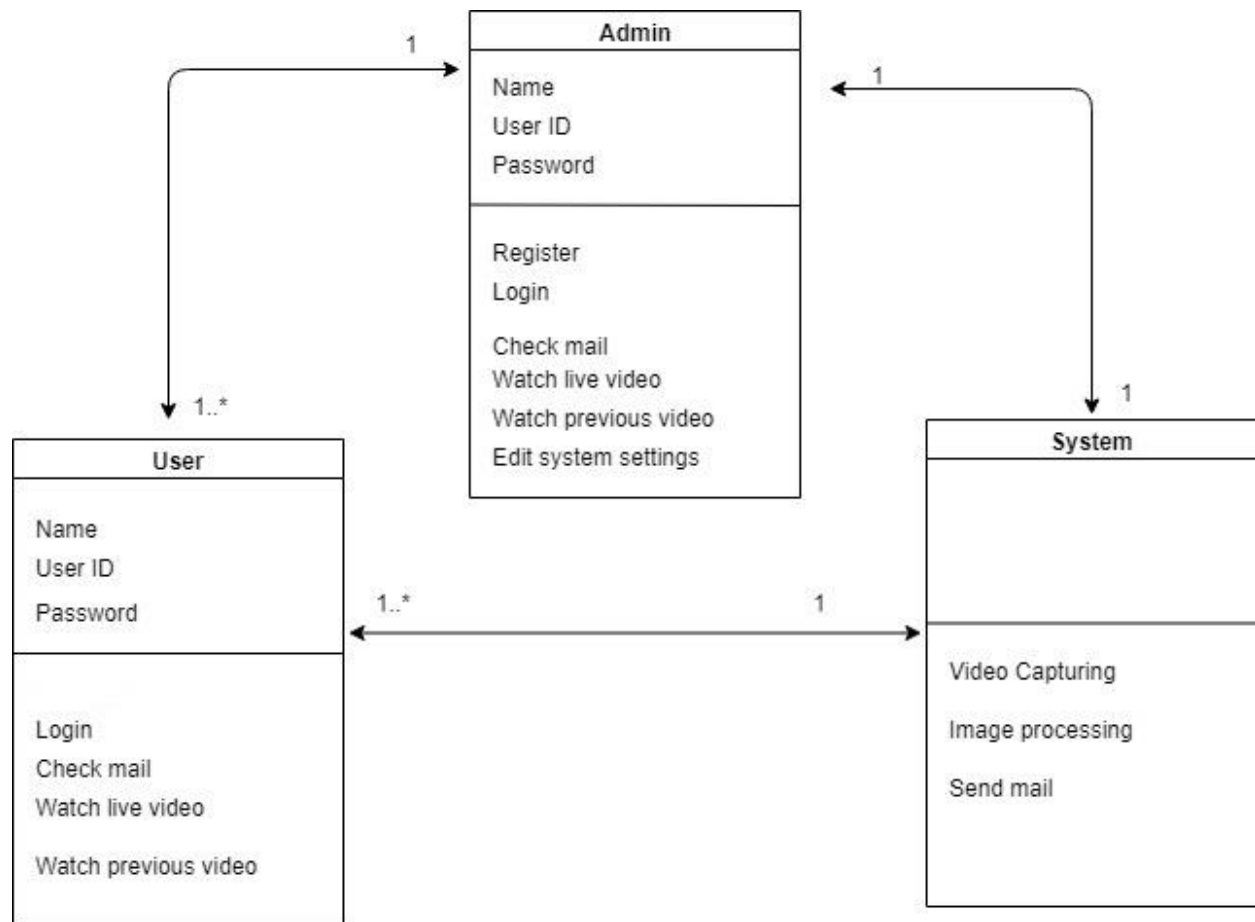


Figure 3.5 : Class Diagram

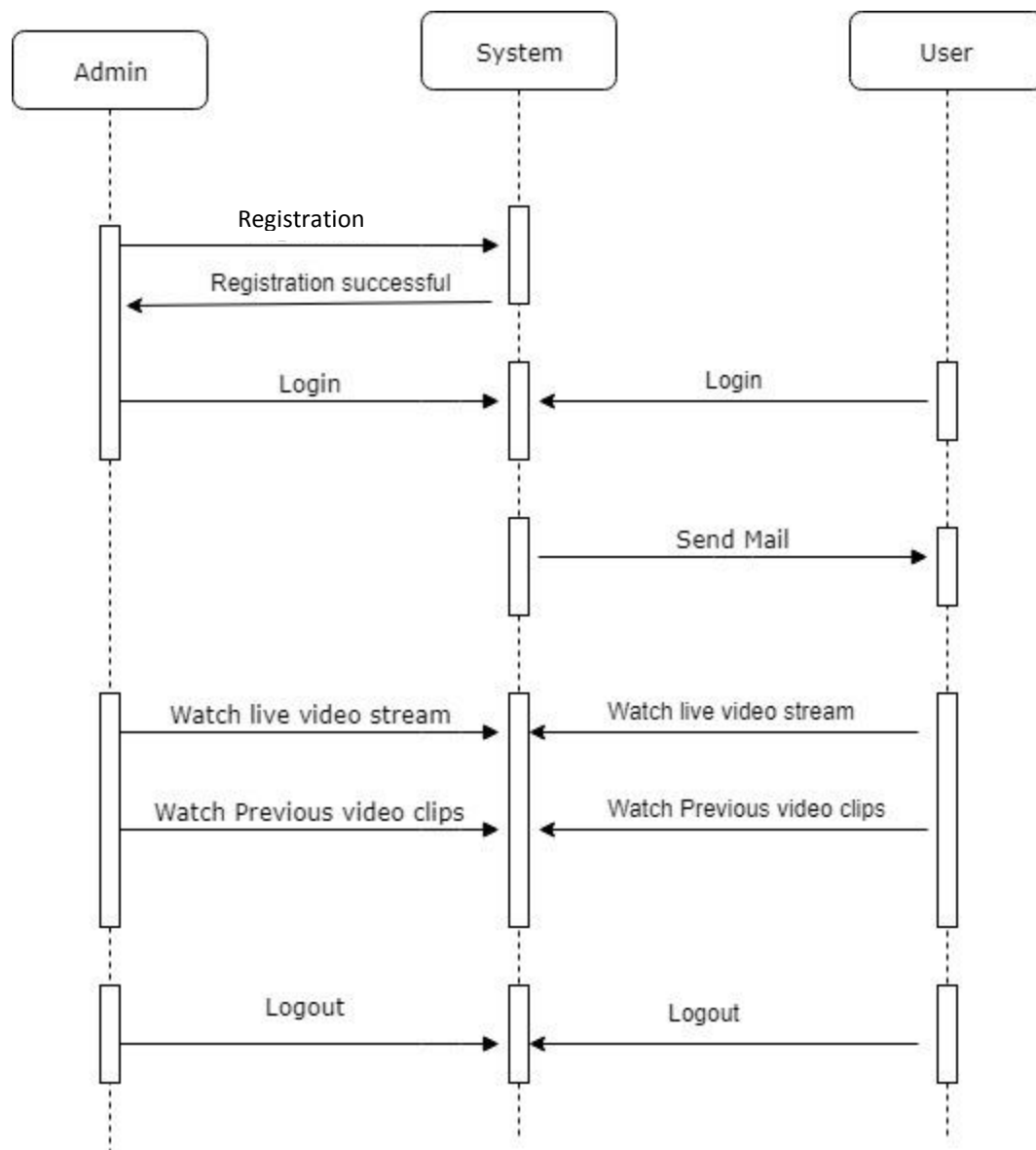


Figure 3.6 : Sequence Diagram

3.9 Internal functions of the system

When we consider the internal functions of the system, first the VGA camera feeds video output to the computer at a speed of 30 frames per second. Then Ubuntu VM grabs all the frames and few processing happen inside the system. First step is that OpenCV algorithm is used to filter out some noise in each frame. Then RGB mode frame is converted to the Gray mode. This part is very important to save the processing power of the CPU. Next step is background reduction.

It is an inbuilt algorithm of the OpenCV which is used easily to compare the current frame with previous one. Then system will check if new object is found on a current frame. If that object is greater than threshold pixel value, then system will generate an email to the configured recipient. After one E-mail was generated, system waits a few seconds to generate another E-mail.

3.10 Coding for the important steps used in OpenCV

Following are the most important functions used in OpenCV for the project.

- Image Acquisition – OpenCV supports several types of image capturing like camera input feed, prerecorded video stream or directory path.

Taking Input from a directory path –

```
img = cv2.imread('C:\Users\USER\Desktop\image.jpg',0)
```

Capturing input from a Video stream -

```
cap = cv2.VideoCapture(0)
```

(Video capture(0) means inbuilt camera. If it is external camera, it should be Video capture(1))

- Save the image –

```
cv2.imwrite('image.jpg', equ)
```

- Display image –

```
cv2.imshow('Input', img)
```

- Convert image from BGR color scheme to Grey or BGR to HSV -

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

- Background Reduction –

```
fpbg = cv2.bgsegm.createBackgroundSubtractorMOG();
fgmask = fpbg.apply(gray);
```

3.11 Chapter Summary

This chapter discussed the details of the overall system design and the way the approach was implemented. Furthermore, it clearly justified why the RAD design methodology was used for the development. Other than that, important algorithms of image processing phase were discussed together with some sample codes of important steps followed in the project. Diagrams presented are really useful to get a clear idea about the project to anyone who has even less technical knowledge.

Chapter 4

Testing and Evaluation

4.1 Chapter Overview

In this chapter, author will outline the testing and evaluation procedures that are being performed throughout the development process and when examining the final version of the application. This also includes a discussion of the expected and desired test results, which define the criteria for success of the project. Further, it describes the possibilities to overcome the failure situations and what methods will be used to overcome such situations. This chapter is concluded with an overview of the testing results with respect to probability of intruder detection rates and performance (speed) of the application in different testing environments.

4.2 Testing Methods

There are two basic testing methods available, Black box testing and White box testing. The White box testing focuses on the internal structure of the code. The white box tester (most often the developer of the code) looks into the coding of the system that has been implemented to look for any errors by writing test cases. Here, the inputs are chosen specifically to determine an appropriate output and the result can be evaluated accordingly. Black box testing is usually described as focusing on testing functional requirements. Knowledge of the systems code and programming knowledge is not required. The test cases are built around what the software is supposed to do. Valid and invalid inputs are taken to determine whether the correct output is given by the system or not. This is mostly used for validation of the system. The black box testing technique here was applied to test the stated functionality which ensures that the right product has been built. This type of testing aims to uncover errors which include interface errors, performance errors as well as initialization and termination errors.

4.3 Description of the Testing procedures

Testing took place throughout the different stages of the application development in order to ensure adequate performance and recognition rates when implementing new functionalities. The testing includes checks for both the performance of the image processing module and the e-mail sending module. Some informal tests regarding the correctness of the output of application were done for each completed unit and module integration step. This dealt with code coverage criteria in particular, in order to ensure that all statements in the code have been executed and tested for validity, and do not contain any bugs or errors.

It has to be noted that accessing a camera (development system's built-in laptop camera) is not possible when recording the video. Therefore, IPCamera (Third party Software) is used to capture images while recording the video with a mobile phone camera. (This is only used to capture the frames for the system). Due to the movement of the camera, light levels of the room, shadows, winds and sudden movements of intruder, the results are also expected to be less accurate on the live system. It was tried to compensate for these circumstances by keeping a fixed location on mobile phone, controlling the light levels, eliminating winds and using inbuilt filtering mechanisms to reduce the noise. By carrying out testing using mobile phone camera, other applications run at the same time could be stopped. On the completion of each and every part of the system, the prototype is tested for compliance with the requirements defined.

4.4 Test Cases

Test No	Description	Desired Outcome	Actual Outcome	Result (Pass / Fail)
1	Initialize Camera	Display Video frames on RGB Format	Display a Live video as frames	Pass
2	Image Conversion	Convert RGB Image into Gray Color Mode	Display a Live video in Gray Color	Pass
3	Image Subtraction	Compare the current frame with previous	Display the variance by using white pixels,	Pass (Filtering noise and

		frame and display if any variance.	otherwise display totally black frame	manually controlling the environment)
4	Save the frame	Check the Background subtracted frame and count the white pixel value. If white pixels are more than threshold, save the image.	If white pixels are more than threshold value, save the image.	Pass (Sometimes fail due to noises and movement of objects)
5	Capture and save the video	Capture all the frames and save it on hard drive	Save the video in phone memory	Pass
6	E-Mail generation	Generate an E-mail when intruder is detected and attached a frame captured with intruder.	Send e-mails when intruder detected.	Pass
7	Change the lighting (Check the background subtracted frames with different light levels)	Correctly identify an intruder	If light levels are too low or too high, Intruder identification is incorrect.	Fail
8	Sudden Movement of intruder	Find intruder on several continuous frames and save each and every frame	Cannot clearly identify an intruder on original frame. It is shown as a noise	Fail
9	Test with different sizes of object movements	If object size is below the threshold size do not detect as an intruder	Most of the time it did not detect as intruder, Some noisy situations it detect as intruder.	Pass (Due to high noise it sometimes Fail)

10	Test with different distances from the main door of the room	If the distance is changed, need to calibrate intruders size (Threshold value) in order to correctly identify an intruder	When distance is low, Small objects were detected as intruders and when distance is high some actual intruders were not detected.	Pass (Time to time need to be calibrated threshold pixel value according to the distance)
11	Check the system to generate e-mails when intruder moves continuously inside the room	E-mail only generates after the desired time interval and previously stored image changed with new image.	E-mail sends within time intervals.	Pass
12	Exit the System	When user pressed “q” or Control + C buttons together, exit the system.	Exit the system after pressing “q” or Control + C	Pass

Table 4.1: Test Cases

4.5 System Performance Evaluation

The overall execution speed of the intruder detection application is a major aspect in evaluation of the system. The first phase, compared each and every video frame with its previous one and identified if any new object appeared in the new frame. One of the OpenCV's inbuilt algorithm (Background Subtraction) was used for completing this phase. OpenCV has implemented three such algorithms and all of them were used to find the best one for the application.

➤ BackgroundSubtractorMOG –

This is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It was introduced in the paper “An improved adaptive background mixture model for real-time tracking with shadow detection” by P. KadewTraKuPong and R. Bowden [7] in 2001.

It uses a method to model each background pixel by a mixture of K Gaussian distributions ($K = 3$ to 5). The weights of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static.



Figure 4.1 : Original Frame Before Background Subtraction [7 , fig .1]

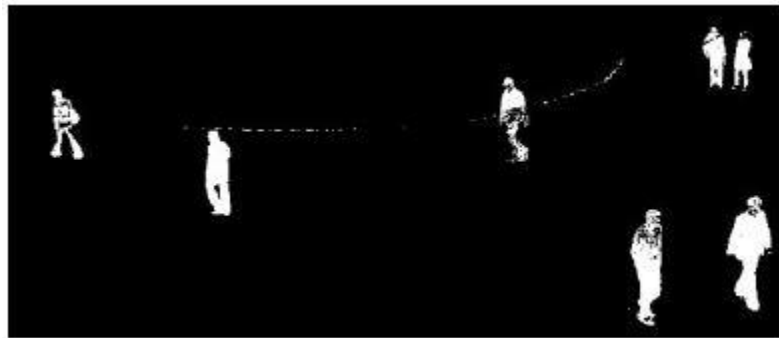


Figure 4.2 : Results of Background SubtractorMOG [7 , fig .2]

➤ BackgroundSubtractorMOG2 -

This is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It is based on two papers by Z.Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction" [7] in 2004 and "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction" in 2006. One important feature of this algorithm is that it selects the appropriate number of Gaussian distribution for each pixel. It provides better adaptability to varying scenes due to illumination changes etc.



Figure 4.3 : Results of Background SubtractorMOG2 [7 , fig .3]

➤ BackgroundSubtractorGMG –

This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation. It was introduced by Andrew B. Godbehere, Akihiro Matsukawa, Ken Goldberg [7] in their paper “Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation” in 2012. It uses first few (120 by default) frames for background modelling. It employs probabilistic foreground segmentation algorithm that identifies possible foreground objects using Bayesian inference. The estimates are adaptive; newer observations are more heavily weighted than old observations to accommodate variable illumination. Several morphological filtering operations like closing and opening are done to remove unwanted noise. It will give a black window during first few frames.



Figure 4.4 : Results of Background SubtractorGMG [7 , fig. 4]

The evaluation results of all three methods are included in appendix section. Comparing the outputs given by each of the algorithm, BackgroundSubtractorMOG algorithm is used for this application.

The quality of the image capturing camera is significant to final output of the system. Usually the blurred images cause the noise in the video frames and those noise reduces the quality of the processed results. Further Image capturing speed of the camera is very important for the performance of the application. It can vary from 50 frames per second (fps) – 7 fps. If it is of high value, application needs high processing power to analyze the frames. If the value is very low, it also negatively affects the quality of the system. Therefore 12 fps – 25 fps speeds are used to run the application.

The next major phase of the application is to automatically generate an E-mail attached with intruder detected frame and send it to the desired receiver. In this task pre-determined time interval is introduced to stop sending unnecessary emails. Therefore, only one email is generated within this time interval and disregarded all the other frames with intruder detected. If emails were continuously generated, it could be identified as a virus by Gmail servers. Therefore, it is really important to send emails in timely manner.

4.6 Chapter Summary

This chapter outlined the testing and evaluation procedures that were carried out throughout the development process to ensure the quality and performance of the produced application. Both the efficiency and effectiveness of the intruder detection module were evaluated, along with standard software testing procedures that were carried out in order to check the correctness of the application. Based on the system performance it was noted that optimizing the application with desired objectives is key for achieving an efficient implementation.

Chapter 5

Conclusion

5.1 Introduction

This chapter will present an overall summary and a discussion of the project and the final output with respect to the project and the difficulties encountered. It further discusses the wrong turns and success stories, challenging situations that were gone through in different stages of this project.

The present study discussed and combined methods taken from a number of different research disciplines, such as image processing, noise reduction and software development. This makes it a valuable piece of work that, while providing an extensive review of the different areas and their application for image processing tasks, may also function as a starting point for further exploration of the aforementioned topics. While not all of the main objectives were achieved, the significant amount of research carried out for this study, as well as the clearly laid out methodologies, the structured system design and the exploration of different approaches to the problem demonstrate the general feasibility of the task based on the chosen solution. It is strongly encouraged to carry out further research based on the conclusions drawn from this work.

5.2 The evaluation of the results of the project

When considering the overall project, the methodology that was planned to be working is almost working as expected. But there are unsuccessful results as well. The main cause for the unsuccessful results are the deviations of the factors like light, color, environment factors and noise from the normal conditions towards some extreme. Still in some scenarios the errors can be minimized by adjusting filtering mechanisms and using some built in libraries.

5.3 Things achieved via this project

This project is mostly related to image processing and video analysis. This subject area is a newly introduced and vastly developing area. This subject is a somewhat new area for me and I haven't had any academic background related to image processing. Therefore, I studied this area from basics and referred several OpenCV manuals, articles, YouTube videos and similar projects related to image processing. This work helped me a lot to apply that knowledge and to understand how those theoretical algorithms, concepts are applied in a real situation. Although the video analysis is a new research area and I did not possess sufficient knowledge of how things are done related to video analysis, still I could gain much familiarity and awareness on video analysis related concepts and theories like Background Subtraction, Motion detection, etc.

At a time when I was looking for suitable technology, the computer vision library called OpenCV and Matlab were quite useful during the research. But Matlab is a commercial software and its license fee is very high. Therefore, freely available OpenCV library was used.

5.4 Problems encountered in the project

The main difficulties were to find detailed documentation on OpenCV functionalities related to Video analysis and find suitable libraries for required functions. Most of the time, error results were occurred due to light variations, speed of object movement, distance and environmental conditions. When it comes to testing since this has to be tested with camera and the object movement, sometimes I could not handle the testing by myself alone and therefore, I obtained the assistance of a third person to move the object while parameters were adjusted. Initially I used an inbuilt laptop camera as the video capturing device and it is very difficult to handle due to space and other factors. Then I attempted to use CCTV camera, but unfortunately camera drivers did not properly support OpenCV. Camera configuration also was a difficult task and it did not give positive results. Finally, mobile phone camera was used and it was really easy to handle due to portability.

5.5 Outcomes of the Project

This project aimed at developing an intruder detection system which could be mainly used as a home surveillance system. The project has a distinctive edge in terms of time and efficiency when compared with similar systems discussed here. That is, in an ordinary surveillance system, in case of an incident, all the recorded videos should be manually viewed in order to find the intrusion. This consumes lot of time. In contrast, in the present system, an immediate detection can be obtained because of the email that is generated for the user giving such information as time and image of intrusion. Further, the system has another advantage as it is developed to minimize false alarms. This is achieved through adjusting the threshold pixel value of the image of the intruder. The user can also watch the intrusion through a live video as a follow up and this is another outcome of the project.

5.6 Future work

In order to make the system's output even more useful and accurate, the Artificial Intelligence features can be included in the project. This could be achieved by integrating the Facial Recognition, and Full body recognition modules provided by the OpenCV library. After detecting the intruder, it could be further analyzed by face database. Then authorized persons can be easily identified in the system. Eventually, it would be an appropriate next step to research on how this project can be commercialized for the use of day to day life. This methodology promises a very efficient implementation of intruder detection procedures on real-time image processing on these platforms.

The performance of the processing can be further improved by using parallel computing concepts in to the system. Current implementation uses sequential execution and it should be time consuming. Therefore, OpenMP, MPI and other parallel computing techniques can be applied to improve the performance.

Currently the implemented version is capable of dealing with a video stream coming from a fixed camera which cannot change the orientation. So the functionality can be further improved to facilitate the object tracking capabilities by adding automatic zooming capabilities to the system even when the camera's angle is automatically changing.

References

1. Chiang, A. et al. "Intruder Detection System", BSc thesis, School of Eng. and Applied Sc., Department of Elec. And Systems Eng., Washinton Univ., St.Lous, May. 2014.
Accessed on: Jan.12,2018. [Online]. Available :
https://ese.wustl.edu/ContentFiles/Research/UndergraduateResearch/CompletedProjects/WebPages/sp14/JonathanAllenSiwei/ESE498Report_Chiang_Chua_Su.pdf
2. Panchal, V. , Dimble, H. , Nikam, S. , and Kamatji, S. , "Border Security :Intruder Human Detection using Image processing", International Journal of Innovative science and Research technology, vol 3, Issue 5, pp.309-311 , May 2011.
3. Tamaraw, N. , and Yoshiname, A. , "A Simple intruder detection using a web camera and image processing", Otsuma journal of social information studies, vol.23, pp.147-154 , 2014.
4. T.Ahmed et al. "Automated intruder detection system from image sequences using minimum volume sets", International journal of communication networks and information security, vol.4, No 1, April 2012.
5. Naveenkumar, M. , and Vadivail, A. , "OpenCV for Computer vision applications", Proceedings of National conference on Big data and Cloud computing, pp.52-56 , March 2015.
6. A. Mordvintsev, A.Rahman K. , "Introduction to OpenCV-Python Tutorials" , Feb. 2018.
Accessed on : Jan 9, 2019. [Online] Available :
https://docs.opencv.org/3.4.1/d0/de3/tutorial_py_intro.html
7. OpenCV dev team , "OpenCV 3.00 –dev documentation – OpenCV-Python Tutorials – Video Analysis" , Nov. 2014. Accessed on : Jan 17, 2019. [Online] Available :
https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html

Appendix A : Python Source Code

```
import sys
import numpy as np
import cv2
import os.path
import sys
import pandas as pd #date & time

from PyQt5 import QtCore, QtWidgets
from PyQt5.QtWidgets import QMainWindow, QWidget, QLabel, QLineEdit, QSlider, QPushButton, QGridLayout,
QVBoxLayout, QApplication
from PyQt5.QtCore import QSize, Qt
import urllib.request
# from PyQt5.QtGui import QIcon, QPixmap
from PyQt5.QtGui import QPixmap, QPainter, QColor, QPen, QIcon, QPixmap

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders

ip = None
ThresholdVal = 500
url = None
lowMargeT_level = 100
maxMargeT_level = 50000
setT_level = 500
skipcount = 200
schek = 0

def savelmage(ip):
    print('Connecting...' + ip)
    fgbg = cv2.bgsegm.createBackgroundSubtractorMOG();
    url = 'http://' + ip + ':8080/shot.jpg'
    #url = 'http://test1:testing@192.168.8.100:8080/shot.jpg'

    global ThresholdVal
    global schek

    while True:
        imgRes= urllib.request.urlopen(url)
        imgNp= np.array( bytearray (imgRes.read()),dtype=np.uint8)
        img=cv2.imdecode(imgNp, -1)
```

```

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
fgmask = fgbg.apply(gray)
cv2.imshow('diff_frame', fgmask);
cv2.imshow('Original',img)

cv2.namedWindow('Original',cv2.WINDOW_NORMAL)
cv2.resizeWindow('Original', 640,360)

cv2.namedWindow('diff_frame',cv2.WINDOW_NORMAL)
cv2.resizeWindow('diff_frame', 640,360)

n_white_pix = np.sum(fgmask == 255)

if (schek != 0 and schek < 200) :
    schek = schek + 1
    print('Waiting!!' , schek)

if ((n_white_pix > ThresholdVal) and (schek == 0 or schek == 200)):
    storeImage(img)

cv2.waitKey(10)

if cv2.waitKey(30) & 0xFF == ord('q'):
    system.quit()
else:
    print('ERROR!')
    system.quit()

class MainWindow(QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        # super().__init__()

        ip = None
        global ThresholdVal
        global lowMargeT_level
        global maxMargeT_level
        global setT_leve
        url = None

        # Set window background color
        self.setAutoFillBackground(True)
        p = self.palette()
        p.setColor(self.backgroundRole(), QColor(0, 153, 153))
        self.setPalette(p)

        self.setMinimumSize(QSize(420, 440)) #320 ,140
        self.setMaximumSize(QSize(420, 440))
        self.setWindowTitle("Intelligent Security System")

```

```

sld = QSlider(Qt.Horizontal, self)
sld.setGeometry(28, 315, 357, 30)

self.label_img = QLabel(self)
pixmap = QPixmap('icone_1.png')
self.label_img.setPixmap(pixmap)
self.label_img.move(150, 10)
self.label_img.resize(120, 120)

self.nameLabe2 = QLabel(self)
self.nameLabe2.setText('ADJUST THE THRESHOLD LEVEL')
self.nameLabe2.move(7,295)
self.nameLabe2.resize(190, 10)

self.nameLabe3 = QLabel(self)
self.nameLabe3.setText('ENTER YOUR CAMERA IP HERE')
self.nameLabe3.move(132,115)
self.nameLabe3.resize(190, 120)

self.line = QLineEdit(self)
self.line.move(110, 190)
self.line.resize(200, 32)

self.line.setText('Enter your IP')
self.line.setText('192.168.1.1')

self.lowMargine = QLabel(self)
self.lowMargine.setText(str(lowMargineT_level))
self.lowMargine.move(3,260)
self.lowMargine.resize(30, 130)

self.lowMargine = QLabel(self)
self.lowMargine.setText(str(maxMargineT_level))
self.lowMargine.move(389,265)
self.lowMargine.resize(190, 120)

self.Tval = QLabel(self)
self.Tval.setText('THRESHOLD LEVEL : ' + str(setT_level))
self.Tval.move(140,350)
self.Tval.resize(190, 10)

self.display = QLabel(self)
self.display.setText(str(maxMargineT_level))

self.display.move(389,265)
self.display.resize(190, 120)

pybutton = QPushButton('OK', self)
pybutton.clicked.connect(self.clickMethod)
pybutton.resize(200,32)
pybutton.move(110, 230)

```

```

pybutton2 = QPushButton('EXIT', self)
pybutton2.clicked.connect(self.close)
pybutton2.resize(60,32)
pybutton2.move(355, 405)

#sroll_1
sld.setRange(lowMargineT_level,maxMargineT_level)
sld.setValue(setT_level)
sld.setTickPosition(QSlider.TicksBelow)
sld.setTickInterval(100)
sld.valueChanged[int].connect(self.valuechange)

def valuechange(self,value):
    print('New ThresholdVal Level : ', value)
    global ThresholdVal
    ThresholdVal = value
    self.Tval.setText(str(ThresholdVal))
    self.Tval.setText('THRESHOLD LEVEL : ' + str(ThresholdVal))
    #saveImage(ip)
    if ip is not None:
        print(ip)

def clickMethod(self):
    ip = self.line.text()
    if ip != 'Enter your IP' :
        print('Logging...')
        print(ip)
        saveImage(ip)
    else :
        print('Set IP First')

def RNamefn():
    rNamefn = pd.datetime.now()
    rename = str(rNamefn.day) + str(rNamefn.month) + str(rNamefn.year) + '_' + str(rNamefn.hour) +
str(rNamefn.minute) + str(rNamefn.second)
    print(rename)
    return(rename)

def storeImage(img) :
    global schek
    schek == skipCal()
    print("point",schek)
    RName = str(RNamefn()) + '.jpg'
    cv2.imwrite(RName, img)
    sendMail(RName, img)
    print(ThresholdVal)

def skipCal():
    global schek

```

```

schek = 1
print('aaaaaaaaaaaa =', schek)
return(schek)

```

```

def sendMail(RName, img):
    fromaddr = 'secupdatenofifymit@gmail.com'
    toaddr = 'amalchaminda81@gmail.com'
    print(RName, 'dddd')

    # instance of MIMEMultipart
    msg = MIMEMultipart()

    # storing the senders email address
    msg['From'] = fromaddr

    # storing the receivers email address
    msg['To'] = toaddr

    # storing the subject
    msg['Subject'] = 'Security Update'

    # string to store the body of the mail
    body = 'Security cam found object'

    # attach the body with the msg instance
    msg.attach(MIMEText(body, 'plain'))

    # open the file to be sent
    filename = RName
    attachment = open(filename, "rb")

    # instance of MIMEBase and named as p
    p = MIMEBase('application', 'octet-stream')

    # To change the payload into encoded form
    p.set_payload((attachment).read())

    # encode into base64
    encoders.encode_base64(p)

    p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

    # attach the instance 'p' to instance 'msg'
    msg.attach(p)

    # creates SMTP session
    s = smtplib.SMTP('smtp.gmail.com', 587)

    # start TLS for security
    s.starttls()

```

```
# Authentication
s.login(fromaddr, 'project2019')

# Converts the Multipart msg into a string
text = msg.as_string()

# sending the mail
s.sendmail(fromaddr, toaddr, text)

# terminating the session
s.quit()

if __name__ == "__main__":

    app = QtWidgets.QApplication([])
    mainWin = MainWindow()
    mainWin.show()
    sys.exit( app.exec_() )

    cap.release()
    cv2.destroyAllWindows()
```

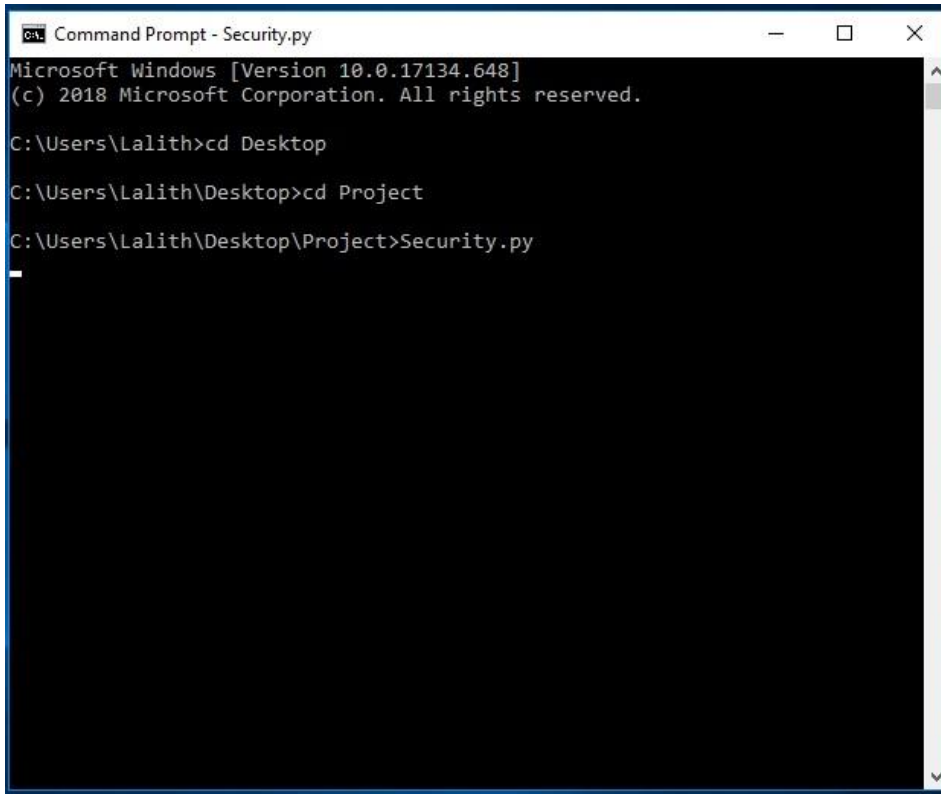

Appendix B : User Manual

In Windows operating system, first you have to open the Command prompt. Then change the working directory to the python program directory.

e.g.: - Currently python program is stored in Project folder of the Desktop.
(cd Desktop/Project)

Then type filename.py in command prompt and press Enter.

e.g.: - Security.py



```
Command Prompt - Security.py
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Lalith>cd Desktop
C:\Users\Lalith\Desktop>cd Project
C:\Users\Lalith\Desktop\Project>Security.py
_
```

Figure 7.1 : Command prompt

Then following screen will appear on the screen. Now you have to enter the Camera IP Address and set the threshold level of the intruder. Finally Press the 'OK 'button.



Figure 7.2 : User Interface Of the Project

Then you have three different screens.

- Command prompt screen:
This screen shows all the entered commands. If intruder is detected, then its details (e.g.: Saved file name, Email sending message, etc.)
- Original Frame screen:
This screen shows live video stream.
- Difference Screen:
This screen shows difference between current frame and previous frame.

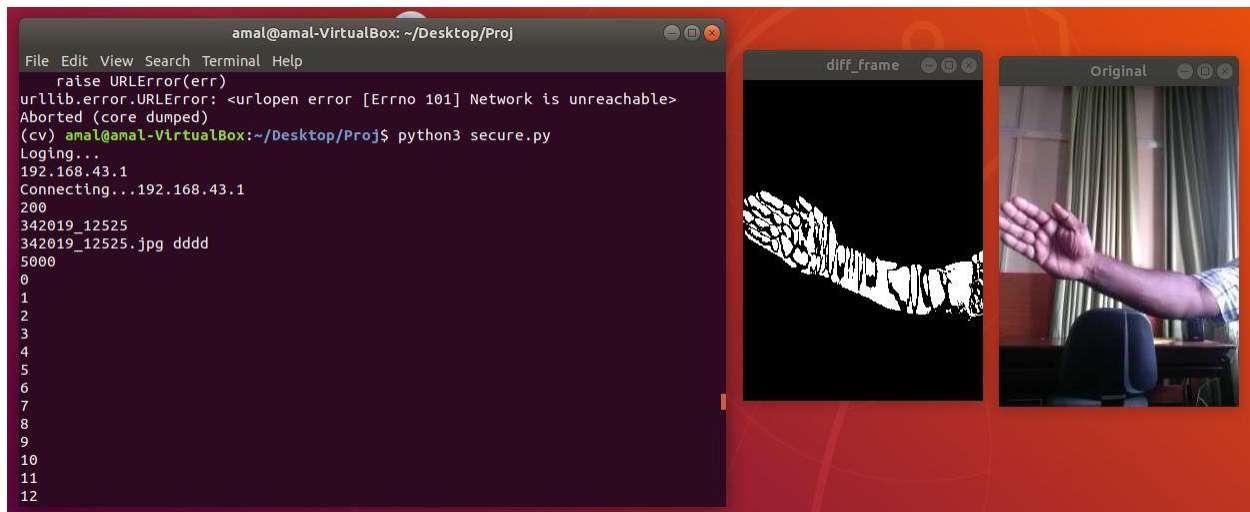


Figure 7.3 : Intruder Detection 1

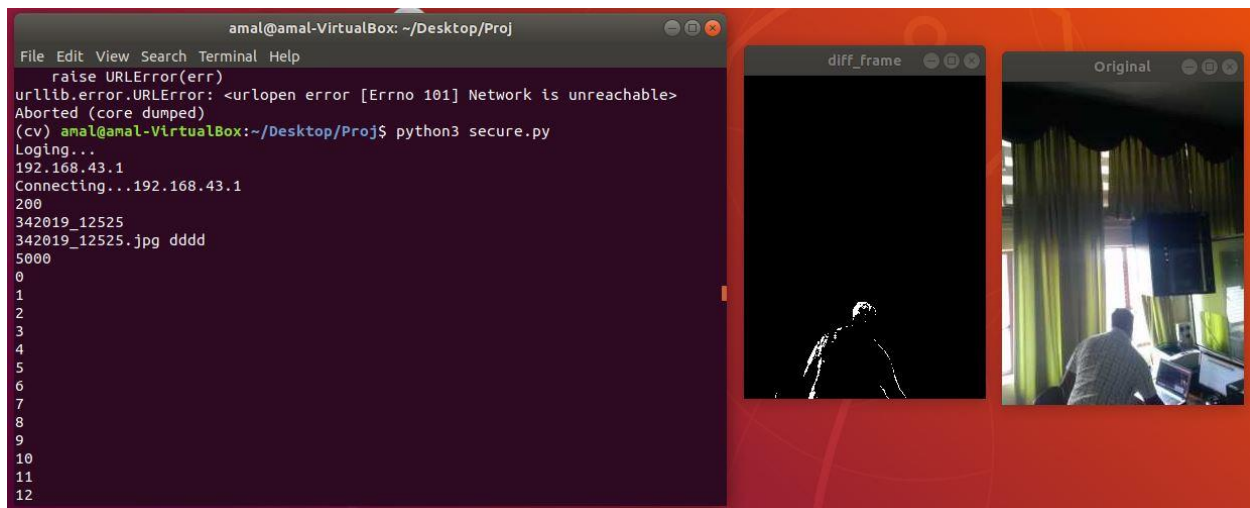


Figure 7.4 : Intruder Detection 2

If intruder is detected and its pixel value is greater than threshold value, then an e-mail is sent to the pre-configured user or users. It can be seen as follows.

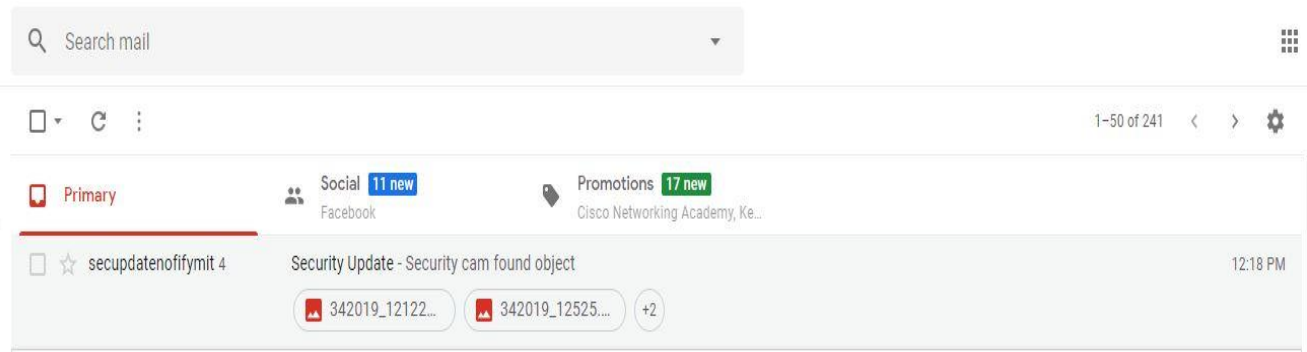


Figure 7.5 : Received Email

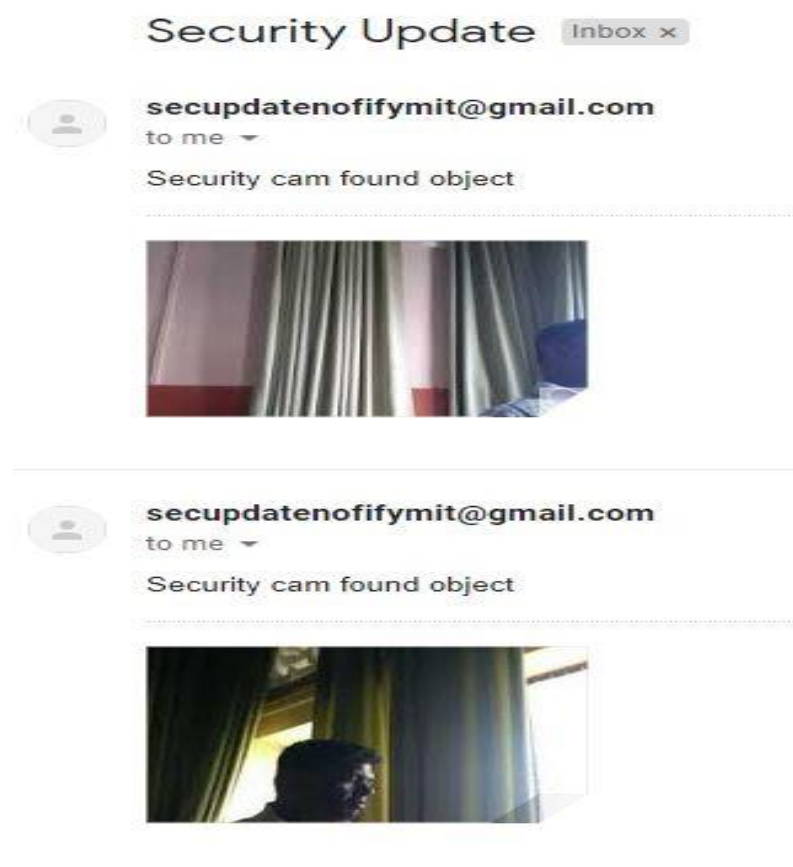


Figure 7.6 : Attachment of the Received Email