4.234Table of Survey Listtable.caption.12

B.Sc. in Computer Science and Engineering Thesis

A Smart Sniper Assistance System Using Image Processing for Enhancing Capability of A Sniper

Submitted by

Brazab Nayak
201414089

Mehreen Hoque
201414098

Md. Mahfujul Islam
201414112

Supervised by

Lt. Col. Amir Hossain Mollah

Associate Professor

Department Of Computer Science & Engineering

Military Institute of Science & Technology

Department of Computer Science and Engineering

Military Institute of Science and Technology

December 2017

CERTIFICATION

This thesis paper titled **”A Smart Sniper Assistance System Using Image Processing for Enhancing Capability of A Sniper”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in January 2017.

Group Members:

1. Brazab Nayak
2. Mehreen Hoque
3. Md. Mahfujul Islam

Supervisor:

____

Lt. Col. Amir Hossain Mollah
Associate Professor, CSE Department
Military Institute of Science & Technology

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper, titled, **"A Smart Sniper Assistance System Using Image Processing for Enhancing Capability of A Sniper"**,is the outcome of the investigation and research carried out by the following students under the supervision of Lt. Col. Amir Hossain Mollah, Associate Professor, CSE Department, Military Institute of Science & Technology.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

____

Brazab Nayak
201414089

____

Mehreen Hoque
201414098

____

Md. Mahfujul Islam
201414112

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor, **Lt. Col. Amir Hossain Mollah**, Associate Professor, CSE Department, Military Institute of Science & Technology, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advice throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2017 .

1. Brazab Nayak

2. Mehreen Hoque

3. Md. Mahfujul Islam

ABSTRACT

A sniper is a person who waits for a specific target to shoot or stalk to gather information.Waiting for the appearance of target for a long time can be tiring for him.To reduce this pressure, our proposed system will automatically identify target and notify him.In this paper,an efficient and cost-consuming system is proposed that will help the sniper to identify the object in a comfortable and technological way.The whole mechanism of this system consists of: Identification of changes in a specific background,comparing a specific change with the previous background.Image processing is mainly used to build this system.It detects a change by measuring the sensitivity of the specific contour area in the image.This measurement is performed by following our modified algorithm.

Index Terms : Sniper, Raspberry-Pi, Python, Image Processing, Target, Cost-consuming, Surveillance System, Automated System, Camera, Pixels, Contour Area

Contents

List of Figures

List of Tables

LIST OF ABBREVIATION

ATR	: Automatic Target Recognition
FPS	: Frame Per Second
HCI	: Human Computer Interface

CHAPTER 1
INTRODUCTION

1.1 Overview

When a sniper needs to focus on a target to be appeared for long time,it can cause a tiresome situation for him.Some cases can also occur such that a sniper may not be able to notice the target because of some kind of hectic situations.According to these issues,this topic has caught attention of many researchers so that any mechanism or system can help them to reduce their troubles.There is a possible solution and method of implementation given in this thesis paper which can be beneficial to a sniper.We have proposed a mechanism using image processing which will help them as an assistant to identify an object for a specific background.We have modified an algorithm and also build a hardware prototype to implement this system.According to the modified algorithm,this system will check if it is day time or night time.In hardware prototype,there will be two cameras which will help to simulate the algorithm. If it is day time, then the system will switch to the first camera.Otherwise it will switch to the second camera.Then the video frame in the camera will be initialized.It will grab a specific frame from the video.According to the frame,the system will measure if any change is occurred or not.In this system,whenever change will be noticed,the system will assume that there is a movement in the background area and it will notify the sniper through text messages.

1.2 Basic Terms

Sniper:

Sniper is known as a marksman.A sniper usually operates by himself.But he can also operates with a team.The main task of a sniper is to maintain close visual contact with the enemy.After getting the contact,he shoots the enemy from a safe position where no one is capable of detecting him.Generally a sniper is a specialist on using rifles and optics which are of high precision.

Sniper Rifle:

A sniper rifle is a long range tactical precision rifle designed for the military and law enforcement for the purposes of antipersonnel, neutralizing materiel and surveillance. The modern sniper rifle is a portable shoulder fired weapon system with an action choice between bolt-action or semi-automatic, fitted with a telescopic sight for high accuracy and chambered for a military center fire cartridge.

Figure 1.1: Snippet of A Modern Sniper Rifle

Challenging Conditions For Snipers:

Snipers shoot targets that are far away. Because the bullet shot by a sniper has to travel a very long distance, the sniper has to be very careful when taking aim. There are three main challenging things that have an effect on the accuracy of the shot:

(i) The target can be in moving position such as it can be seen in walking or running also.A sniper has to think about the position position of the target during the time of shooting because a target might move in a rapid way.
(ii) According to the rules of gravity,the force of it will affects the bullet.Then the bullet will drop towards the ground.When the target will be close to the sniper,this rule does n't effect on the result of accuracy.But when the target is in long distance,then the sniper should keep in mind that how high he has to shoot because it will make up the distance of the falling bullet as it flies.
(iii) Wind can affect a bullet.It can cause a bullet to travel and also to blow it from the target.In this case,sniper must think about the direction and also keep the information about the strength of the wind.In this case sometimes,sniper has to shoot to the left or right of the target.In this way,sniper can make up for the wind.

Auto Target Recognition Process: Automatic target recognition (ATR), is the ability for an algorithm or device to recognize targets or objects based on data obtained from sensors. Target recognition was initially done by using an audible representation of the received signal, where a trained operator who would decipher that sound to classify the target illuminated by the radar.

1.3 Main Idea & Objective

Our proposed system is named as a smart sniper assistance system. We will built a hardware prototype along with software prototype.It is a mechanism that has used Raspberry Pi as a hardware device and Python as a programming language.The reason for using python because Raspberry pi because Python is used as a real time version and it is very user-friendly.Whenever sniper feels tiresome,this device will work as an assistant of a sniper.It will notify to him via text-messages that whether any change in the targeted area has occurred or not.

1.4 Thesis Organization

In **chapter 2**, the discussion regarding the research in this field and similar works and researches regarding to our proposed system is held.

In **chapter 3**, we have described our methodology of the proposed system.

In **chapter 4**, results and analysis of our proposed system is discussed in a specific way.

At last in **chapter 5**, we concluded our thesis work.

CHAPTER 2
LITERATURE REVIEW

2.1 Image Processing

Image processing is one kind of method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Digital image processing focuses on two major tasks-

(i) Improvement of pictorial information for human interpretation.

(ii) Processing of image data for storage, transmission and representation for autonomous machine perception.
Digital Image is a two-dimensional function x and y are spatial coordinates the amplitude of f is called intensity or gray level at the point (x, y).

2.2 Methods of Image Processing

There are mainly two types of image processing.They are:
(i) Analog Image Processing Technique.
(ii) Digital Image Processing Technique.

2.2.1 Analog Image Processing

Analog or visual techniques of image processing can be used for the hard copies like print-outs and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques.

2.2.2 Digital Image Processing

Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies.

2.3 Evolution of Image Processing

Evolution of Image Processing has gone through a journey. In short,the glimpse of the whole evolution is given below:

2.3.1 Early 1920s

One of the first applications of digital imaging was in the news-paper industry. Initially-

- This method was the Bart-lane cable picture transmission service.

- Images were transferred by submarine cable between London and New York

- Pictures were coded for cable transfer and reconstructed at the receiving end on a telegraph printer.

- Sent by submarine cable between London and New York, the transportation time was reduced to less than three hours from more than a week.

2.3.2 Mid to late 1920s

Improvements to the Bart-lane system resulted in higher quality images

(i) New reproduction processes based on photographic techniques

(ii) Increased number of tones in reproduced images

2.3.3 1960s

Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing.

2.3.4 1970s

In 1970s Digital image processing begins to be used in medical applications. In 1979 Sir Godfrey N. Hounsfield and Prof. Allan M. Cormack share the Nobel Prize in medicine for the invention of tomography, the technology behind Computerized Axial Tomography (CAT) scans.

2.3.5 1980s to the Present

The use of digital image processing techniques has exploded and they are now used for all kinds of tasks in all kinds of areas such as-

(i) Image enhancement/restoration

(ii) Artistic effects

(iii) Medical Visualization

(iv) Industrial inspection

(v) Human computer interfaces (HCI)

2.4 Field of Image Processing

2.4.1 The Hubble Telescope

(i) Launched in 1990 the Hubble telescope can take images of very distant objects.

(ii) However, an incorrect mirror made many of Hubbles images useless.

(iii) Image processing techniques were used to fix this.

2.4.2 Artistic Effects

Artistic effects are used to make images more visually appealing, to add special effects and to make composite images.

Human Computer Interaction (HCI)

Human Computer Interaction (HCI) researches the design and use of computer technology, focusing particularly on the interfaces between people (users) and computers. Researchers in the field of HCI both observe the ways in which humans interact with computers and design technologies that let humans interact with computers in novel ways such as-

(i) Trying to make human computer interfaces more natural.

(ii) Face recognition.

(iii) Gesture recognition.

2.4.3 Face Morphing

Morphing refers to a computer technique used for graphics and in films, in which one image is gradually transformed into another image without individual changes being noticeable in the process. Morphing is a special effect in motion pictures and animations that changes (or morphs) one image or shape into another through a seamless transition.

2.4.4 Bio-metric Fingerprint Recognition

Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of bio-metric used to identify individuals and verify their identity.

2.4.5 Iris Recognition

Iris recognition bio-metric systems apply mathematical pattern-recognition techniques to images of the irises of an individual's eyes. Iris recognition is an automated method of bio-metric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex random patterns are unique, stable, and can be seen from some distance.

2.4.6 Handwriting Recognition

Handwriting Recognition using Neural Network in Image Processing. This methods are based on extraction of a small set of parameters, and pure neural methods, in which the neural net-work is fed with raw image data.

2.5 Stages of Image Processing

Image Acquisition is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves pre processing, such as scaling etc.

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Example: changing brightness and contrast etc.

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

2.6 Image Recognition

Image Recognition is the process that assigns a label in a particular image. Representation and description almost always follow the output of a segmentation stage which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself.

Image processing algorithms are basically developed to overcome different problems. Some of these include image restoration,image enhancement,image segmentation and the list goes on.

2.7 Image Denition

An image is a visual representation of something. In information technology,the term has several usages:

1.An image is a picture that has been created or copied and stored in electronic form. An image can be described in terms of vector graphics or raster graphics. An image stored in raster form is sometimes called a bitmap. An image map is a le containing in formation that associates different locations on a specied image with hyper text links.

2.An image is a section of random access memory(RAM) that has been copied to another memory or storage location.

A digital image is a representation of a two-dimensional image as a nite set of digital values,called picture elements or pixels. Pixel values typically represent gray levels, colors, heights, opacity etc. Digitization implies that a digital image is an approximation of a real scene.

2.7.1 Binary Image

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

2.7.2 Gray Scale Image

A gray scale or gray scale digital image is an image in which the value of each pixel is a single sample,that is, it carries only intensity information.

2.7.3 Color Image

A (digital) color image is a digital image that includes color information for each pixel. For visually acceptable results, it is necessary (and almost sufcient)to provide three samples (color channels) for each pixel, which are interpreted as coordinates in some color space. The RGB color space is commonly used in computer displays, but other spaces such as YCbCr, HSV, and are often used in other contexts.

2.8 Pixels

Pixel is the elements of a digital image. The pixel (a word invented from ”picture element”) is the basic unit of programmable color on a computer display or in a computer image. In digital imaging, a pixel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device; so it is the smallest controllable element of a picture represented on the screen. The address of a pixel corresponds to its physical coordinates.

2.9 Contour Area

Contour Area is returning the area inside a contour. It should be closed, otherwise, it will consider itself that the area is closed. As for example, if it is a line it will consider the area inside the contour defined by the line if the first point and the last point are linked.

2.9.1 Contour Properties

There are some properties that affects in the calculation of the contour area. They are:

1. Aspect Ratio

It is the ratio of width to height of bounding rectangle area of the object.

Aspect Ratio = $\frac{Width}{Height}$

2. Extent

Extent is the ratio of contour area to bounding rectangle area.

Extent =$\frac{Object\ Area}{Bounding\ Rectangle\ Area}$

3. Solidity

Solidity is the ratio of contour area to its convex hull area.

Solidity = $\frac{Contour\ Area}{Convex\ Hull\ Area}$.

4. Equivalent Diameter

Equivalent Diameter is the diameter of the circle whose area is same as the contour area.

Equivalent Diameter = $\sqrt{\frac{4 \times Contour\ Area}{\pi}}$

5. Orientation

Orientation is the angle at which object is directed. Following method also gives the Major Axis and Minor Axis lengths.

6. Mask and Pixel Points

In some cases, we may need all the points which comprises that object. Finding these points two methods can be applied.One using Numpy functions, next one using OpenCV function.Their implementation gives the same result but with a slight difference.

Numpy gives coordinates in (row, column) format.
OpenCV gives coordinates in (x,y) format.

7. Maximum Value, Minimum Value and their locations

We can find these parameters using a mask image.

8. Mean Color or Mean Intensity

We can find the average color of an object. Or it can be average intensity of the object in gray scale mode. We again use the same mask to do it.

9. Extreme Points

Extreme Points means topmost, bottom most, rightmost and leftmost points of the object.

2.10 Video Analysis in OpenCV

By following some techniques and algorithms in OpenCV, we can find a particular objects and track them in a videos.Thos formulae are:

2.10.1 Mean Shift Algorithm

To use mean shift in OpenCV, first we need to setup the target, find its histogram so that we can back our project the target on each frame for calculation of mean shift. We also need to provide initial location of window. For histogram, only Hue is considered here. Also, to avoid false values due to low light, low light values are discarded using cv2.inRange() function.

2.10.2 Cam Shift Algorithm

It is almost same as mean shift, but it returns a rotated rectangle (that is our result) and box parameters (used to be passed as search window in next iteration).

2.10.3 Optical Flow

Optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second.Two types of optical flow can be operated in OpenCV.

Lucas-Kanade Optical Flow

OpenCV provides all these in a single function,cv2.calcOpticalFlowPyrLK() Here, we create a simple application which tracks some points in a video. To decide the points, we use cv2.goodFeaturesToTrack(). We take the first frame, detect some Shi-Tomasi corner points in it, then we track those points in an iterative way using Lucas-Kanade optical flow. For the function cv2.calcOpticalFlowPyrLK() we pass the previous frame, previous points and next frame. It returns next points along with some status numbers which has a value of 1 if next point is found, else zero. We pass these next points in an iterative way as previous points in next step.

Dense Optical Flow

Lucas-Kanade method computes optical flow for a sparse feature set (in our example, corners detected using Shi-Tomasi algorithm). OpenCV provides another algorithm to find the dense optical flow. It computes the optical flow for all the points in the frame.

2.10.4 Background Subtraction

Background subtraction is a major pre-processing steps in many vision based applications.If we have an image of background alone,we can subtract the new image from the background. You get the foreground objects alone. But in most of the cases, we may not have such an image, so we need to extract the background from whatever images we have. It become more complicated when there is shadow appears. Since shadow is also moving, simple subtraction will mark that also as foreground. It complicates things.

In case of these analysis,there are many sub genre found in background subtraction method.They are:

Background Subtractor MOG

It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. While coding, we need to create a background object using the function,
cv2.createBackgroundSubtractorMOG(). It has some optional parameters like length of history, number of gaussian mixtures, threshold etc. It is all set to some default values. Then inside the video loop, use backgroundsubtractor.apply() method to get the foreground mask.

Background Subtractor MOG2

It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm.As in previous case, we have to create a background subtractor object. Here,we have an option of selecting whether shadow to be detected or not. If detectShadows = True (which is so by default), it detects and marks shadows, but decreases the speed. Shadows will be marked in gray color.

Background Subtractor GMG

This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation.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.It would be better to apply morphological opening to the result to remove the noises.

2.11 Related Work

A limited number of research has been conducted in this area.This section will provide a brief review of the related work.

1. Some research work has been conducted to develop intelligent mechanism for helping a sniper such as auto target detection of humanoid and auto shooting described in [1]. It is mainly a design approach proposed in this paper.In this approach a new defense mechanism is proposed using image processing. It has been implemented which can acquire targets at long distances with high precision in various weather conditions. According to the paper,the mechanism consists of two modules: targeting unit and firing unit. The targeting unit has a micro controller along with sensors to detect the presence of humans in the targeted area and processes various atmospheric parameters electronically. LASER pointer is used to determine if the desired target is hit or missed by tracking the path of the bullet and determining the point of intersection with the LASER. The targeting unit has special mechanism which can position the gun automatically once the parameters are obtained.

2. In [2] an adaptive Butter-worth high pass filter is used.It is used under a sea-sky complex background so that it can detect a small target.This detection process can be held by the calculation of the weighted information entropy which is of different infrared image.In this process,the cutoff frequency of the filter is able to change in an adaptive way.It is a robust small target detection method according to the experimental result.

3. The paper [3] is about the current state of the art which is in video-based target detection.It also includes an analysis of background adaption techniques.This paper mainly focuses on the LOTS which means Le-high Omni directional Tracking System and its components.It also includes multi background modeling in an adaptive way.A novel approach to spatio-temporal grouping which is known as quasi-connected components is also included here.This paper also emphasizes on camouflaged and adversarial target detection under the

extreme sensitivity.

4. A new defence mechanism method which uses image processing implemented in paper [4].This system is basically focused on finding out the best target tracking technology.It is proposed to use to overcome the foreign attacks which normally happens at the country border.This whole system is divided into two modules:Control unit and Humanoid unit.The control unit consists of a laptop which implements GUI(Graphical User Interface) using MATLAB.The humanoid unit is responsible for receiving signal from the control unit.Image processing technique processes the control signal.The humanoid, works in three modes-
(1) Auto Targeting and Auto Shooting (Fully Autonomous). (2)Auto Targeting and Manual Shooting.(Semi Autonomous). (3)Manual Targeting and Manual S hooting (User Oriented).

5. [5] represents a general problem of signal detection.It is in the background of unknown Gaussian noise.It uses the techniques of statistical hypothesis testing.The system derives ratio decision rules and evaluates its performance in both noise only and signal+noise only.

6.Paper [6] describes a system which is omni-directional.This paper also discusses some of LOTS'(Lehigh Omni-directional Tracking System) components and their unique features.This paper gives combined result regarding to limited visibility distance and target visibility.The approach of this system can give a wide field of view and also can be regarded as a natural application for omni-directional VSAM (video-surveillance and monitoring).

7.In [7],the problem of ATR(Automatic Targer Recognition)is given and some techniques to solve this problem is discussed.Mainly this paper emphasizes on algorithmic and implementation approaches.Usage of multi sensors in this case is also discussed.

In sum, the earlier work mainly focuses on different techniques to detect the targets. To the best of our knowledge, the system described in closely matches with our system whereas our system is a better version of it which will provide cost-reducing mechanism and more efficiency.

CHAPTER 3
METHODOLOGY

3.1 Features of Our Proposed System

Basically our system has some features which has made it unique and easy to implement.Those features are:

1. **Notification Alert:**
   If the sysem observes a slight change in the selected frame,it will notify the sniper that a change has occurred.
2. **Wide Range:**
   Our proposed system can cover upto 30 feet range.Vision range can be broaden if we modify our system with better zooming camera.
3. **Cost-reducing:**
   Though it is not the cheapest,but we have reduced cost in all possible way so that it can be affordable.
4. **Text message alert:**
   Sniper will get text messages of notification in their cell phones.It is because in case if the sniper does some other works then he can miss target.
5. **Fault Tolerance System:**
   This system has provided fault tolerant solution to improve hardware redundancy.

3.2 Component of the system

For designing a prototype we have used following components:

i) Raspberry Pi(Model B+):
The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It

is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything our expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

It has 512MB RAM, four USB ports, 40 GPIO pins, and an Ethernet port. In February 2015, it was superseded by the Pi 2 Model B, the second generation of the Raspberry Pi. The Pi 2 shares many specs with the Pi 1 B+, and originally used a 900MHz quad-core ARM Cortex-A7 CPU and has 1GB RAM.

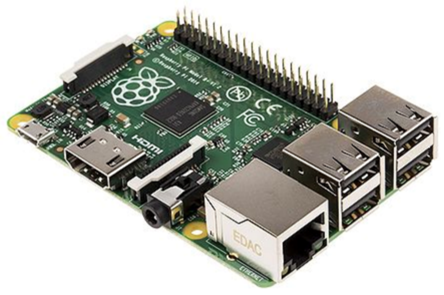


Figure 3.1: Raspberry Pi

ii) IR Camera:
An infrared camera is a non-contact device that detects infrared energy (heat) and converts it into an electronic signal, which is then processed to produce a thermal image on a video monitor and perform temperature calculations.A thermo graphic camera (also called an infrared camera or thermal imaging camera) is a device that forms an image using infrared radiation, similar to a common camera that forms an image using visible light. Their use is called thermo graphy.

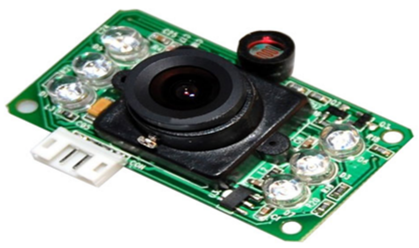


Figure 3.2: Raspberry Pi

iii) Raspberry Pi Camera v2:

The Raspberry Pi Camera v2 is the new official camera board released by the Raspberry Pi Foundation. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSi interface, designed especially for interfacing to cameras. The v2 Camera Module has a Sony IMX 219 8-megapixel sensor (compared to the 5-megapixel Omni-Vision OV5647 sensor of the original camera). The Camera Module can be used to take high-definition video, as well as stills photographs. The camera works with all models of Raspberry Pi 1, 2, and 3.

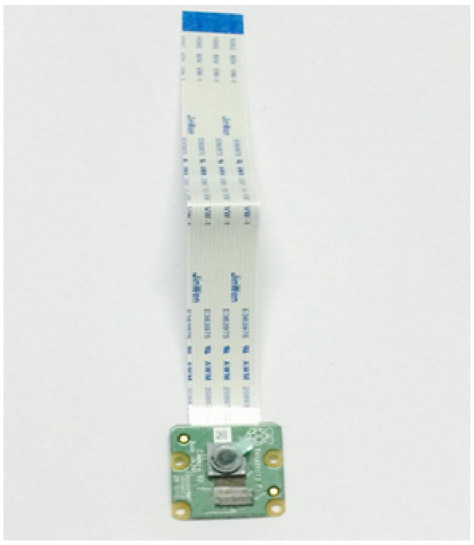


Figure 3.3: Raspberry Pi

3.3 Structure of Our Proposed System

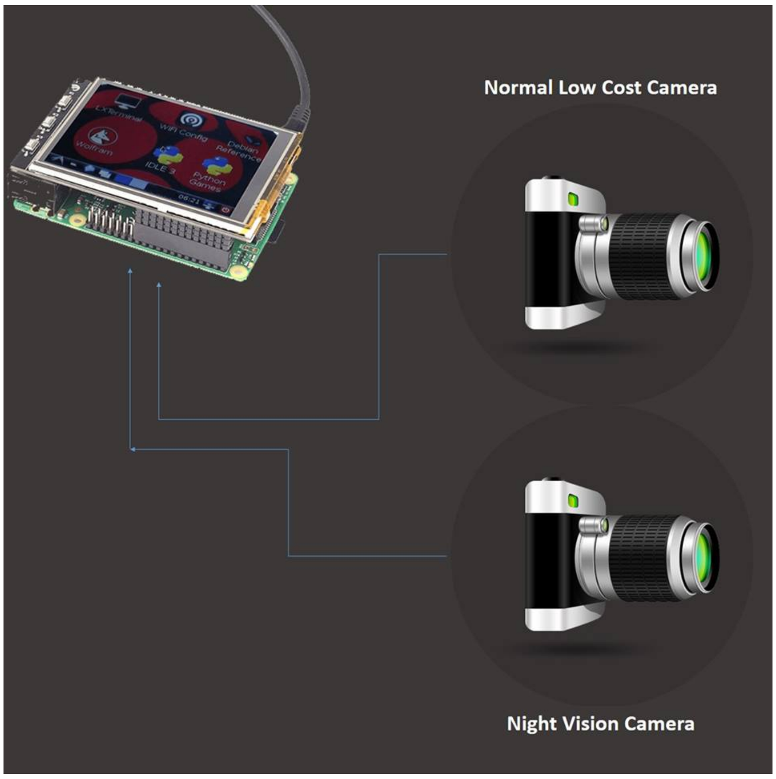


Figure 3.4: Prototype of Our System

In our prototype,there will be two cameras attached in the raspberry device.One camera will work as a day light version and another one will work as a night vision sense.OpenCV library will be used in order to maintain the software part of the program.

3.4 Algorithm

According to the algorithm,this system will check if it daylight or night vision.If it is comparative to daylight then,the system will switch to the first camera.Otherwise it will switch to the 2nd camera.Then it will initialize the video frame and at first it's number will be 0.Then it will grab a frame from the video.If no frame cannot be grabbed then the program will be terminated.Otherwise the frame number will be incremented by 1.Then the contour area will be calculated.If the contour area is greater than sensitivity,the number will be incremented by 2.Then this number again will be incremented by 1.Then the areas will be compared and the contour area will be detected by shape.The change will be saved and notified to the system.

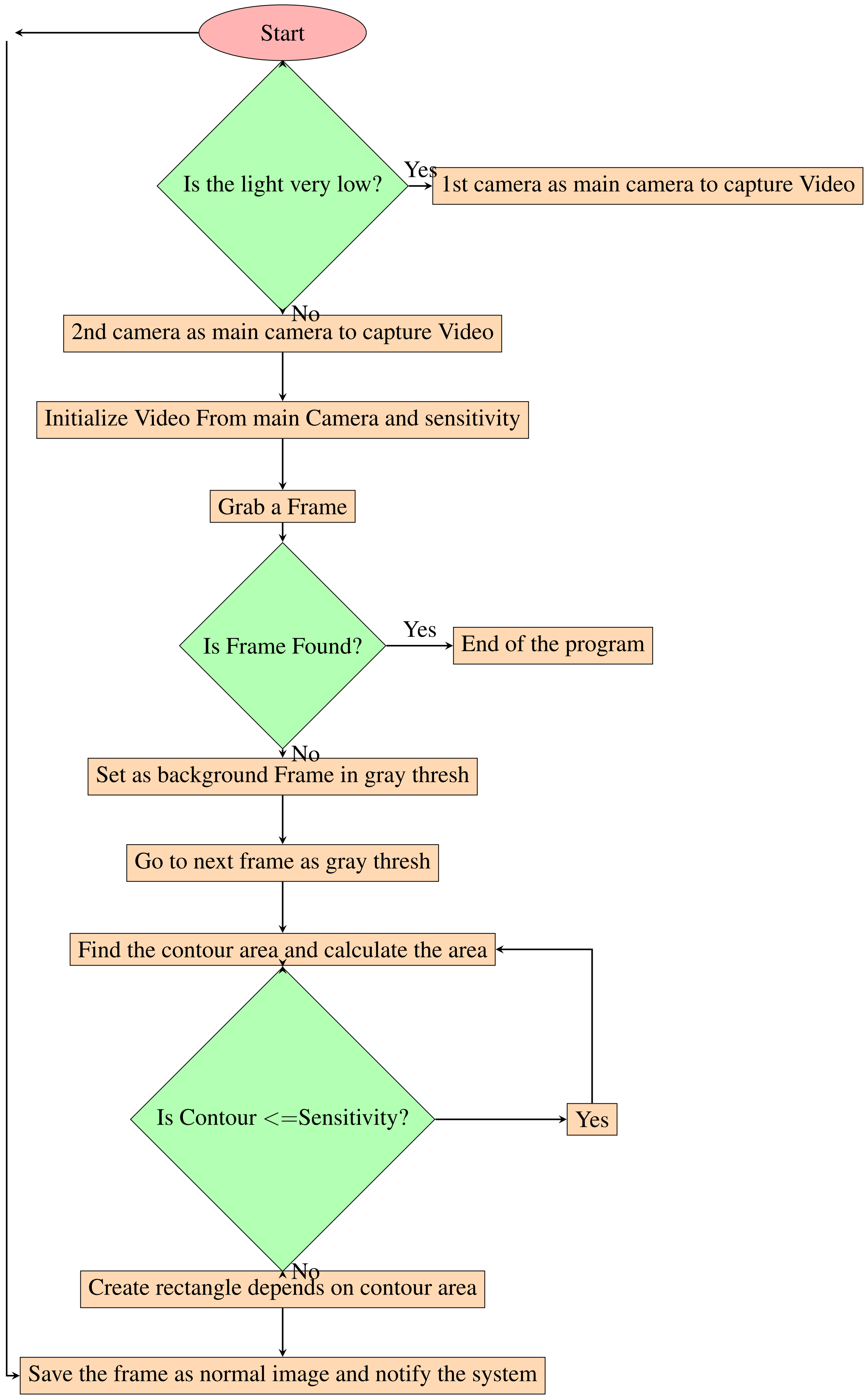
Algorithm 1 Change in Detection Frame

```
While (true)
    Check the daylight
    if (light is low) then
        Switch to 1st camera as the main camera
    else
        Switch to 2nd camera as the main camera
    end if
    Initialize the video frame ← main camera
    Number ← 0
Grab a Frame from the video
If No Frame is grabbed
End of Program
else (Background Frame ← Frame(Number+0))
Test frame ← frame(Number+1)
Contour Area ← Calculate the area of Contour
    if (Contour area < Sensitivity) then
        Background Frame ← Frame (Number+2)
        Test Frame ← Frame(Number+3)
    else
        Detect the contour area by shape
        Save the frame as image and notify the system
    end if
end if
End
```

3.5 Flowchart

A flowchart is a one kind of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a our problem.

In this flow chart,the system will start at first.Then it will check the light of the environment.If the light around the system is low,it will activate the 1st camera which is the night vision camera as the main camera to capture the video.If the light is sufficient or the opposite of low,the system will activate the second camera as the main camera.The second camera is the normal camera which can operate in the daylight also.After the selection of the camera,the camera will initialize the video frame and sensitivity.Then it will grab a frame.After choosing frame,the mechanism will decide whether the system finds the similar frame that is previously grabbed or some changes in the frame is occurred.If it finds the similar frame,then the program will be terminated.But if the system finds any change in the frame,it will set it as the background frame.Then it will find the next frame and identify the changed area which is known as contour area.Then the whole system will calculate the contour area.If the measurement of the contour area is less than or equal to the sensitivity,then the system will ignore the change and will go to the previous step.But if the contour area is greater than the predefined sensitivity,then the system will identify the main changes through a rectangle and save the image.Lastly,after saving the image,the system will notify to the system.



Start
Is the light very low?
Yes
1st camera as main camera to capture Video
No
2nd camera as main camera to capture Video
Initialize Video From main Camera and sensitivity
Grab a Frame
Is Frame Found?
Yes
End of the program
No
Set as background Frame in gray thresh
Go to next frame as gray thresh
Find the contour area and calculate the area
Is Contour <=Sensitivity?
Yes
No
Create rectangle depends on contour area
Save the frame as normal image and notify the system

CHAPTER 4
RESULT AND ANALYSIS

Image Restoration is the operation of taking a corrupt/noisy image and estimating the clean,original image in which the input is an image,such as a photograph or video frame.The output of image processing maybe either an image or a set of characteristics or parameters related to the image.

4.1 Cases

In this proposed system,the whole system functions upon the basis of light.There are two cases can be performed in this mechanism:
1.When the light is low and
2.When the light is sufficient.

4.1.1 When The Light is Low:

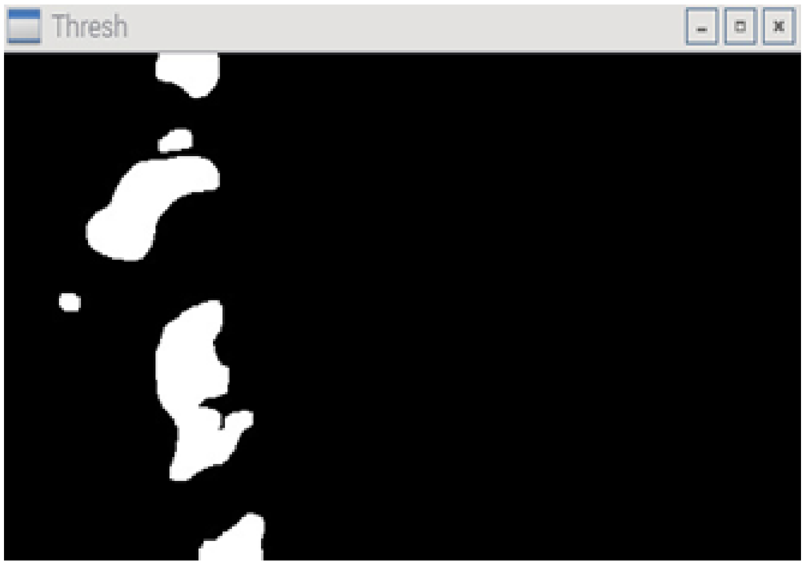


Figure 4.1: Snippet of Night Version Image

When the light is low,the system activates the night vision camera and the whole system works through the gray scale image. In night vision camera,images are normally black and white as we can see in the Figure 5.1. That black area indicates the unchanged background and the white area indicates the change in the background. By calculating these white contour area which is calculated in pixels,we can detect and find the area where the change occurs.

4.1.2 When The Light is Sufficient:

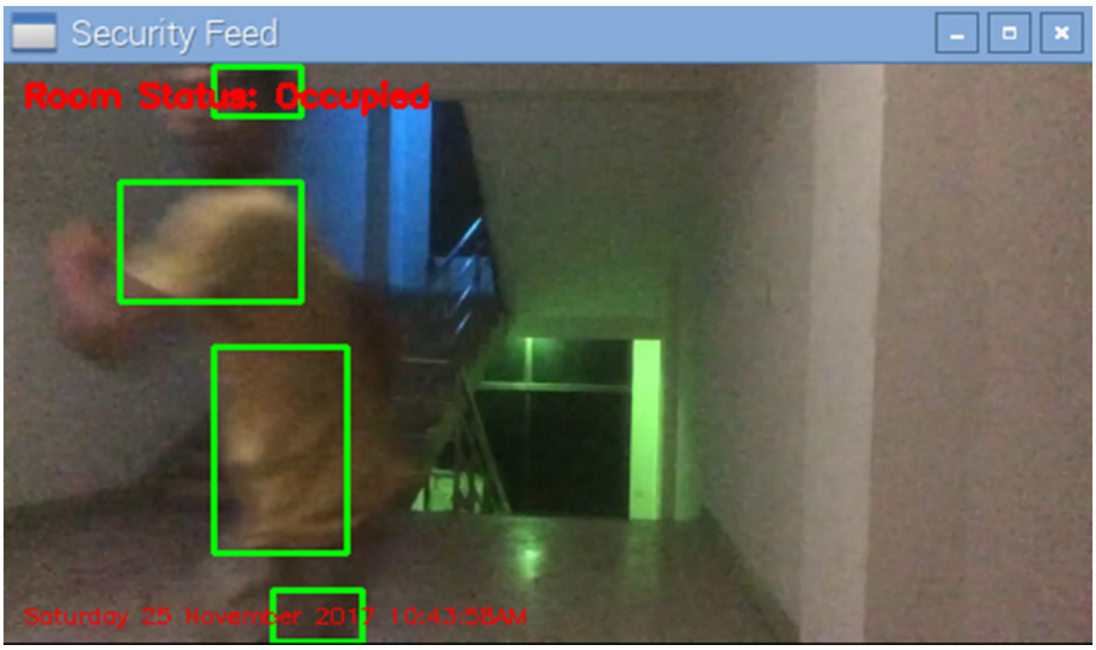


Figure 4.2: Snippet of Image in Normal Light

When the light is normal,the normal camera activates. It can identify the change in colorful pixels. If the given frame and the occurred frame have some different pixels,then the system detect those pixels and calculate the changes. Then it notifies the system.

4.2 Analysis

Cartridge	Ranges
5.5645mm NATO (.223 Remington)	300–500 m
7.6251mm (.308 Winchester)	800-1,000 m
7.6254mmR	800-1,000 m
7 mm Remington Magnum	900-1,100 m
.338 Lapua Magnum	1,300-1,600 m
.50 BMG (12.799mm NATO)	1,500-2,000 m
12.7108mm (Russian)	1,500-2,000 m
14.5114mm (Russian)	1,900-2,300 m

Table 4.1: Table to test captions and labels

We have analyzed various sniper rifles' ranges and cartridges from different datasheets.That analysis has been presented in the table 4.1.

We have conducted a survey among 10 students to experiment about how good and efficient our system is.Information of this survey is presented in table 4.2.

Survey List				
User name	User experience/UI(Out of 10)	Software Efficiency(Out of 10)	Hardware Efficiency(Out of 10)	Remarks
Asibul Islam	5	7	8	Need to use real camera
Amir Hamza	4	7	9	Need to improve UX.
Md. Mahfujul	6	8	8	Not user friendly
Maruf Abdul Matin	5	6	6	Slow system
Tarikul Islam	6	8	9	Need to work fast.
Hasibur Rahman Hridoy	6	9	9	Overall good.
Kivran Bin Moktar	4	8	7	Need to update hardware.
Md Jubaier	7	8	9	nice System.
Md. Mahfulul Islam	7	9	8	need to improve UI.
Adnan Sharif	4	8	10	Overall Good

Table 4.2: Table of Survey List

CHAPTER 5
CONCLUSION

We have built our prototype based on a small range of vision. If it can be done for a wide area such as the border area of a country,then a sniper will be helpful and it will reduce his pressure on physical activity.

5.0.1 Limitations

No system can be operated without system loss or some barriers.In this solution and given prototype,there are some limitations. They are:

(i) We need to be careful to implement under the real-life conditions.There are many small changes can be appeared in the background frame which may need to be considered.But unfortunately our system,couldn't detect the small changes in the background.

(ii) A very high resolution camera is needed in order to take a very clear image.At present,the camera is used in this system,cannot detect an image if the distance gets longer.

(iii) Time rendering problem in case of capturing the frame is one of the vital problem in this system.Let's suppose we capture a video which has 10 FPS duration.Then if any changes occur during 100 nano-second,the system won't notify that particular change.But this problem can be solved if we can use a higher resolution camera.

5.0.2 Future Work

At present,our proposed system can detect changes occurring in the particular frame.In future,we can efficient our algorithm for each type of images.Besides,sometimes,it happens that a target moves very fast and changes this position in a very rapid way.A sniper cannot detect this movement sometimes due to tiresome situation.So,In near future,target tracking algorithm can be used to detect the movement of the target and many path finding algorithm can be used in order to find the path in which the target has moved and changed his position. We hope that our proposed system will be highly strong and powerful helping tool for a sniper at the war field.Moreover, we will work to improve this system in all possible way based on present technology.

REFERENCES

[1] V. Baruah, M. Vyas, and T. Mahanta, “Auto target detection of humanoid and auto shooting using sniper: A design approach,” in Control, Measurement and Instrumentation (CMI), 2016 IEEE First International Conference on. IEEE, 2016, pp. 469–473.

[2] E. J. Kelly, “An adaptive detection algorithm,” IEEE transactions on aerospace and electronic systems, no. 2, pp. 115–127, 1986.

[3] T. Boult, R. Micheals, X. Gao, P. Lewis, C. Power, W. Yin, and A. Erkan, “Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets,” in Visual Surveillance, 1999. Second IEEE Workshop on,(VS’99). IEEE, 1999, pp. 48–55.

[4] T. E. Boult, R. J. Micheals, X. Gao, and M. Eckmann, “Into the woods: Visual surveillance of noncooperative and camouflaged targets in complex outdoor settings,” Proceedings of the IEEE, vol. 89, no. 10, pp. 1382–1402, 2001.

[5] J. A. Shajahan, S. Jain, C. Joseph, G. Keerthipriya, P. K. Raja et al., “Target detecting defence humanoid sniper,” in Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on. IEEE, 2012, pp. 1–6.

[6] T. Boult, R. Micheals, X. Gao, P. Lewis, C. Power,W. Yin, and A. Erkan, “Frame-rate omnidirectional surveillance and tracking of camouflaged and occludedtargets,” inVisual Surveillance, 1999. Second IEEEWorkshop on,(VS’99). IEEE, 1999, pp. 48–55.

[7] B. Bhanu, “Automatic target recognition: State of the artsurvey,”IEEE transactions on aerospace and electronicsystems, no. 4, pp. 364–379, 1986.

[8] S. W. Vetter and E. Leclerc, “Novel aspects of calmodulintarget recognition and activation,”The FEBS Journal,vol. 270, no. 3, pp. 404–414, 2003.

[9] M. W. Roth, “Survey of neural network technologyfor automatic target recognition,”IEEE Transactions onneural networks, vol. 1, no. 1, pp. 28–43, 1990.

[10] K.-N. Chia, H. J. Kimet al., “Configurable computingsolutions for automatic target recognition,” inFPGAs forCustom Computing Machines, 1996. Proceedings. IEEESymposium on. IEEE, 1996, pp. 70–79.

[11] J. A. Ratches, C. Walters, R. G. Buser, and B. Guenther,“Aided and automatic target recognition based upon sen-sory inputs from image forming systems,”IEEE transactions on pattern analysis and machine intelligence,vol. 19, no. 9, pp. 1004–1019, 1997.

[12] D. P. Bartel, “Micrornas: target recognition and regula-tory functions,”cell, vol. 136, no. 2, pp. 215–233, 2009.

[13] N. A. Tanner, Y. Zhang, and T. C. Evans Jr, “Simultane-ous multiple target detection in real-time loop-mediatedisothermal amplification,”Biotechniques, vol. 53, no. 2,pp. 81–89, 2012.

[14] H. Rohling, “Radar cfar thresholding in clutter and mul-tiple target situations,”IEEE transactions on aerospaceand electronic systems, no. 4, pp. 608–621, 1983.

[15] M. R. Morelande, C. M. Kreucher, and K. Kastella, “Abayesian approach to multiple target detection and track-ing,”IEEE Transactions on Signal Processing, vol. 55,no. 5, pp. 1589–1604, 2007.

[16] G. Pulford, “Taxonomy of multiple target tracking meth-ods,”IEE Proceedings-Radar, Sonar and Navigation, vol.152, no. 5, pp. 291–304, 2005.

[17] R. Braunling, R. M. Jensen, and M. A. Gallo, “Acoustictarget detection, tracking, classification, and location ina multiple-target environment,”Peace and Wartime Applications and Technical Issues for Unattended GroundSensors, vol. 3081, no. 1, pp. 57–66, 1997.

[18] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, “Movingtarget classification and tracking from real-time video,”inApplications of Computer Vision, 1998. WACV’98.Proceedings., Fourth IEEE Workshop on.IEEE, 1998,pp. 8–14.

[19] M. P. Ekstrom, “Digital image processing techniques,” 2012.

[20] R. M. Lougheed, “Neighborhood image processing stage for implementing ltering operations,” September 1985.

[21] J. C. Russ and R. P. Woods, “The image processing handbook,” Journal of Computer Assisted Tomography, vol. 19, no. 6, pp. 979–981, 1995.

APPENDIX
Python File

```
python motion-detector.py
python motion-detector.py
```

```
//import the necessary packages import argparse
import datetime
import imutils
import time
import cv2
import os

//day light constant(avg)
valDay=128
// construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add-argument(”-v”, ”–video”, help=”path to the video file”)
ap.add-argument(”-a”, ”–min-area”, type=int, default=500, help=”minimum area size”)
args = vars(ap.parse_args())
```

```
//If the video argument is none, then we are reading from webcam ifargs.get(”video”, None)
is None.
```

Code To Detect The Light Condition :

```
sumCal=0;
(grabbed, frame) = camera.read()
frame=RBG2Gray(frame)

for x in range (0, length.pixel(frame))
for y in range (0,width.pixel(farame)) sumCal=sumCal+pixelValue(x,y);
sumCal=sumCal/y
```

Code To Swift The Camera :

```
if(sumCal¿valDay)
//As main camera as normal vision camera
camera = cv2.VideoCapture(0)
elseif
//As second camera as night vision camera
camera = cv2.VideoCapture(1)
time.sleep(0.25)

//Otherwise, we are reading from a video file
else
camera = cv2.VideoCapture(args[”video”])

// Initialize the first frame in the video stream
firstFrame = None

loop over the frames of the video
while True:
//grab the current frame and initialize the occupied/unoccupied
// text
(grabbed, frame) = camera.read()
text = ”Unoccupied”

//if the frame could not be grabbed, then we have reached the end of the video
if (not grabbed)
break

//Resize the frame, convert it to grayscale, and blur it
frame = imutils.resize(frame, width=500)
gray = cv2.cvtColor(frame, cv2.COLOR-BGR2GRAY)
gray = cv2.GaussianBlur(gray, (21, 21), 0)
```

```
//if the first frame is None, initialize it
if (firstFrame is None:)
firstFrame = gray
continue


//Compute the absolute difference between the current frame and
first frame
frame-Delta = cv2.absdiff(firstFrame, gray)
thresh = cv2.threshold(frameDelta, 25, 255, cv2.THRESH-BINARY)[1]


//Dilate the thresholded image to fill in holes, then find contours on thresholded image
thresh = cv2.dilate(thresh, None, iterations=2)
image,cnts,hierarchy =
cv2.findContours(thresh.copy(),cv2.RETR-EXTERNAL,cv2.CHAIN-APPROX-SIMPLE)


// loop over the contours
for c in cnts:
//if (the contour is too small)
ignore it
if (cv2.contourArea(c) ¡args[”min-area”]:)
continue


//compute the bounding box for the contour, draw it on the frame and update the text
(x, y, w, h) = cv2.boundingRect(c)
cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
//For taking a screen shot
os.system(”scrot”)
text = ”Occupied”


//send sms to notify
//Your Account SID from twilio.com/console
account-sid = ”ACa139730c19aaebdaba8b6294522bf770”
```

```
//Your Auth Token from twilio.com/console


auth-token = ”8a022c9e1e35d866c0d2528e661d21ef”
client = Client(account-sid, auth-token)
message = client.messages.create(to=”+8801939710407”,from-=”+17327068272”,body=”Please
check Your System update!!!!!!”)
print(message.sid)


//Show the frame and record if the user presses a key


cv2.imshow(”Security Feed”, frame)
cv2.imshow(”Thresh”, thresh)

cv2.imshow(”Frame Delta”, frameDelta)
if (user’s need)
key = cv2.waitKey(1)  0xFF


if (the ‘q‘ key is pressed, break from the loop)
if key == ord(”q”):
break


//cleanup the camera and close any open windows
camera.release()
cv2.destroyAllWindows()
```

MeanShift in OpenCV

```
import numpy as np
import cv2

cap = cv2.VideoCapture('slow.flv')

take first frame of the video
ret,frame = cap.read()

setup initial location of window
r,h,c,w = 250,90,400,125
simply hardcoded the values
track-window = (c,r,w,h)


set up the ROI for tracking
roi = frame[r:r+h, c:c+w]
hsv-roi = cv2.cvtColor(frame, cv2.COLOR-BGR2HSV)
mask = cv2.inRange(hsv-roi, np.array((0., 60.,32.)), np.array((180.,255.,255.)))
roi-hist = cv2.calcHist([hsv-roi],[0],mask,[180],[0,180])
cv2.normalize(roi-hist,roi-hist,0,255,cv2.NORM-MINMAX)


Setup the termination criteria, either 10 iteration or move by atleast 1 pt


term-crit = ( cv2.TERM-CRITERIA-EPS — cv2.TERM-CRITERIA-COUNT, 10, 1 )


while(1):
ret ,frame = cap.read()


if ret == True:
hsv = cv2.cvtColor(frame, cv2.COLOR-BGR2HSV)
dst = cv2.calcBackProject([hsv],[0],roi-hist,[0,180],1)


apply meanshift to get the new location
ret, track-window = cv2.meanShift(dst, track-window, term-crit)


Draw it on image
x,y,w,h = track-window
```

```
img2 = cv2.rectangle(frame, (x,y), (x+w,y+h), 255,2)
cv2.imshow('img2',img2)

k = cv2.waitKey(60)  0xff
if k == 27:
break
else:
cv2.imwrite(chr(k)+".jpg",img2)

else:
break

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

CamShift in OpenCV

```
import numpy as np
import cv2

cap = cv2.VideoCapture('slow.flv')

take first frame of the video
ret,frame = cap.read()

setup initial location of window
r,h,c,w = 250,90,400,125
simply hardcoded the values
track-window = (c,r,w,h)

set up the ROI for tracking
roi = frame[r:r+h, c:c+w]
hsv-roi = cv2.cvtColor(frame, cv2.COLOR-BGR2HSV)
mask = cv2.inRange(hsv-roi, np.array((0., 60.,32.)), np.array((180.,255.,255.)))
roi-hist = cv2.calcHist([hsv-roi],[0],mask,[180],[0,180])
cv2.normalize(roi-hist,roi-hist,0,255,cv2.NORM-MINMAX)

Setup the termination criteria, either 10 iteration or move by atleast 1pt term-crit = ( cv2.TERM-
CRITERIA-EPS — cv2.TERM-CRITERIA-COUNT, 10, 1 )

while(1):
ret ,frame = cap.read()

if ret == True:
hsv = cv2.cvtColor(frame, cv2.COLOR-BGR2HSV)
dst = cv2.calcBackProject([hsv],[0],roi-hist,[0,180],1)

apply meanshift to get the new location
ret, track-window = cv2.CamShift(dst, track-window, term-crit)

Draw it on image
```

```
pts = cv2.boxPoints(ret)
pts = np.int0(pts)
img2 = cv2.polylines(frame,[pts],True, 255,2)
cv2.imshow('img2',img2)

k = cv2.waitKey(60)  0xff
if k == 27:
break
else:
cv2.imwrite(chr(k)+".jpg",img2)

else:
break

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

Lucas-Kanade Optical FLow in OpenCV

```
import numpy as np
import cv2

cap = cv2.VideoCapture('slow.flv')

params for ShiTomasi corner detection
feature-params = dict( maxCorners = 100,
qualityLevel = 0.3,
minDistance = 7,
blockSize = 7 )

Parameters for lucas kanade optical flow
lk-params = dict( winSize = (15,15),
maxLevel = 2,
criteria = (cv2.TERM-CRITERIA-EPS —
cv2.TERM-CRITERIA-COUNT, 10, 0.03))

Create some random colors
color = np.random.randint(0,255,(100,3))

Take first frame and find corners in it
ret, old-frame = cap.read()
old-gray = cv2.cvtColor(old-frame, cv2.COLOR-BGR2GRAY)
p0 = cv2.goodFeaturesToTrack(old-gray, mask = None, **feature-params)

Create a mask image for drawing purposes
mask = np.zeros-like(old-frame)

while(1):
ret,frame = cap.read()
frame-gray = cv2.cvtColor(frame, cv2.COLOR-BGR2GRAY)

calculate optical flow
p1, st, err = cv2.calcOpticalFlowPyrLK(old-gray, frame-gray, p0, None, **lk-params)
```

```
Select good points
good-new = p1[st==1]
good-old = p0[st==1]

draw the tracks
for i,(new,old) in enumerate(zip(good-new,good-old)):
a,b = new.ravel()
c,d = old.ravel()
mask = cv2.line(mask, (a,b),(c,d), color[i].tolist(), 2)
frame = cv2.circle(frame,(a,b),5,color[i].tolist(),-1)
img = cv2.add(frame,mask)

cv2.imshow('frame',img)
k = cv2.waitKey(30)  0xff
if k == 27:
break

Now update the previous frame and previous points
old-gray = frame-gray.copy()
p0 = good-new.reshape(-1,1,2)

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

Dense Optical FLow in OpenCV

```
import cv2
import numpy as np
cap = cv2.VideoCapture(”vtest.avi”)

ret, frame1 = cap.read()
prvs = cv2.cvtColor(frame1,cv2.COLOR-BGR2GRAY)
hsv = np.zeros-like(frame1)
hsv[...,1] = 255

while(1):
ret, frame2 = cap.read()
next = cv2.cvtColor(frame2,cv2.COLOR-BGR2GRAY)

flow = cv2.calcOpticalFlowFarneback(prvs,next, None, 0.5, 3, 15, 3, 5, 1.2, 0)

mag, ang = cv2.cartToPolar(flow[...,0], flow[...,1])
hsv[...,0] = ang*180/np.pi/2
hsv[...,2] = cv2.normalize(mag,None,0,255,cv2.NORM-MINMAX)
rgb = cv2.cvtColor(hsv,cv2.COLOR-HSV2BGR)

cv2.imshow(’frame2’,rgb)
k = cv2.waitKey(30)  0xff
if k == 27:
break
elif k == ord(’s’):
cv2.imwrite(’opticalfb.png’,frame2)
cv2.imwrite(’opticalhsv.png’,rgb)
prvs = next

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

Background SubtractorMOG in OpenCV

```
import numpy as np
import cv2


cap = cv2.VideoCapture('vtest.avi')


fgbg = cv2.createBackgroundSubtractorMOG()


while(1):
ret, frame = cap.read()


fgmask = fgbg.apply(frame)


cv2.imshow('frame',fgmask)
k = cv2.waitKey(30)  0xff
if k == 27:
break


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

Background SubtractorMOG2 in OpenCV

```
import numpy as np
import cv2


cap = cv2.VideoCapture('vtest.avi')


fgbg = cv2.createBackgroundSubtractorMOG2()


while(1):
ret, frame = cap.read()


fgmask = fgbg.apply(frame)


cv2.imshow('frame',fgmask)
k = cv2.waitKey(30)  0xff
if k == 27:
break


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

BackgroundSubtractorGMG in OpenCV

```
import numpy as np
import cv2

cap = cv2.VideoCapture('vtest.avi')
kernel = cv2.getStructuringElement(cv2.MORPH-ELLIPSE,(3,3))
fgbg = cv2.createBackgroundSubtractorGMG()

while(1):
ret, frame = cap.read()

fgmask = fgbg.apply(frame)
fgmask = cv2.morphologyEx(fgmask, cv2.MORPH-OPEN, kernel)

cv2.imshow('frame',fgmask)
k = cv2.waitKey(30)  0xff
if k == 27:
break

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

Computing Moments

```
class Moments

public:
Moments();
Moments(double m00, double m10, double m01, double m20, double m11, double m02,
double m30, double m21, double m12, double m03 );
Moments( const CvMoments moments );
operator CvMoments() const;


// spatial moments
double m00, m10, m01, m20, m11, m02, m30, m21, m12, m03;
// central moments
double mu20, mu11, mu02, mu30, mu21, mu12, mu03;
// central normalized moments
double nu20, nu11, nu02, nu30, nu21, nu12, nu03;
```