Suspicious Human-Movement Detection

Supervisor: Dr. Jia Uddin

Amlan Biswas- 13101179

Sadia Chowdhury Ria- 13301064

Zannatul Ferdous – 13301089

Shamiha Nishat Chowdhury – 13301090

Department of Computer Science and Engineering,

BRAC University

Submitted on: 21st August 2017

DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor	Signature of Author
Dr. Jia Uddin	Amlan Biswas
	Sadia Chowdhury Ria
	Zannatul Ferdous
	Shamiha Nishat Chowdhury

ACKNOWLEDGEMENT

We would like to thank Dr. Jia Uddin for agreeing to supervise us with our thesis. His patience and confidence in us has been a source of encouragement and this thesis would not have been possible without the great support, inspiration and influence of him. We believe his dedication to this paper deserves to be reciprocated with great gratitude.

A special thanks to the Thesis committee for taking the time to review and evaluate our thesis as part of our undergraduate program.

TABLE OF CONTENTS

DECLARATIONS i
ACKNWOLEDGEMENTS ii
TABLE OF CONTENTS iii

ABSTRACT **1**

CHAPTER 01: INTRODUCTION

1.1 Motivation 2
1.2 Contributions Summary 2
1.3 Thesis Orientation 3

CHAPTER 02: BACKGROUND INFORMATION

2.1 Suspicious Behavior 4
2.2 Digital Image Processing 4
2.2.1 Coordinate Convention 5
2.2.2 Matrix Form Of Images 7
2.3 Image Segmentation 8
2.3.1 Morphological Image Processing 8
2.3.1.1 Structuring Elements 9
2.3.1.2 Opening 10
2.3.1.3 Closing 11
2.4 Color Conversion 12
2.5 Background Subtraction 12
2.6 Using Frame Differencing 13
2.7 Mean Filter 14

2.8 Running Gaussian Average 15
2.9 Related Works 16

CHAPTER 03: PROPOSED MODEL

3.1 Introduction 18
3.2 Flowchart 19
3.3 Creates System Objects 20
3.3.1 Creates System Objects for Foreground Mask 20
3.3.2 Creates System Objects for Blob Analysis 20

3.4 Read Frame from Video 21
3.5 Detects Objects from Frame 21
3.5.1 Detect Foreground 22
3.5.1.1 Morphologically Open Image 22
3.5.1.1.1 Creating Structural Element 22
3.5.1.1.2 Eroding Gray Scale Image 23
3.5.1.1.3 Dilating Gray Scale Image 23
3.5.1.2 Morphologically Close Image 23
3.5.1.2.1 Creating Structural Element 23
3.5.1.3 Fill up The Holes in Binary Image 24
3.5.2 Blob Analysis 24
3.6 Crop Human Body from the Binary 24
3.7 Training 25
3.7.1 Pre- Processing 25
3.7.2 Converting Binary Image to Column Vector 26
3.8 Processing with Testing Image 26
3.9 Behavior Analysis 27
3.9.1 Euclidian Distance 27

CHAPTER 04: EXPERIMENT RESULT & ANALYSIS

4.1 Setup 29
4.2 Dataset 29
4.3 Result 30
4.4 Comparison 33

CHAPTER 05: CONCLUSION AND FUTURE WORKS

5.1 Conclusion 34
5.2 Future Works 34
Reference 35

ABSTRACT

Video analytics is the method of processing a video, gathering data and analyzing the data for getting domain specific information. This project is about to develop a surveillance security system which use human body movement detection to establish public surveillance security. However, human does not observe suspicious situations, but algorithms that process the captured images and detect suspicious behavior or events. [2].For our experimental evolution we used trained image set and Background Subtraction Algorithm. This system collects images from camera and observe the behavior of human. Common behaviors are not given much attention by the system.

CHAPTER 01

INTRODUCTION

1.1 Motivations

Human face and human behavioral pattern play an important role in person identification. Nowadays with the increasingly growing needs of protection of people and personal properties, video surveillance has become a big concern of everyday life. A consequence of these needs has led to the deployment of cameras almost everywhere. Most current video surveillance systems share one characteristic: they need human operator to constantly watch monitors that show the images captured by the cameras [1]. We can use such algorithms which will capture images and detect suspicious behavior or events [2]. In this case, a video surveillance system which can interpret the scene and automatically recognize suspicious behaviors can play a vital role. The system would then notify operators or users accordingly. Our goal is to develop such method and techniques to make an ideal surveillance system. A large part of this project is devoted to visual analysis of human behavior and detection of suspicious gestures of human in indoor scenarios. In this context, a working system would alert of dangerous situations and improve the personal safety of people living on their own.

1.2 Contributions Summary

The contributions of us in this thesis are summarized as follows:

- We worked with each frame in a video. We took a test video as input and took frame by frame to move on to the next step.
- Separating background scenarios and foreground objects for better object tracking.
- In our thesis, we converted the binary values of the binary images into decimal and created a column vector for training purpose.

1.3 Thesis Orientation

The rest of the thesis is organized as follows:

- Chapter 02 includes the necessary background information regarding the used algorithm.
- Chapter 03 presents our proposed model and its implementation.
- Chapter 04 demonstrates the experimental results and comparison.
- Chapter 05 concludes the thesis and states the future research.

CHAPTER 02

BACKGROUND INFORMATION AND RELATED WORKS

2.1 Suspicious Behavior

Suspicious activity is any observed behavior that could indicate anomalous behavior or crime. Behavior is considered suspicious when it is atypical, out of the ordinary, causes some kind of impairment, or consists of undesirable behavior. It is an assumption that suspicious behavior is a disorder that has a physical cause, specifically that it is related to the physical structure of the brain [12]. Before we mention what is meant by suspicious behavior, we should notice that several keywords were used by many research works [13, 14] to refer to the same notion (unusual, rare, atypical, interesting, suspicious, and anomalous). With Mahadevan et al [15], the suspiciousities are defined as measurements whose probability is below a certain threshold under a normal model. Rare behavior and unusual behavior are not same. There are some differences between them. Rare behavior means that has not been observed before, those that have been seen once are considered as rare but not necessarily suspicious. On the other hand, unusual behaviors can be predicted as suspicious.

2.2 Digital Image Processing

The technology of image processing is concerned with a manipulation of the elements of a picture to enhance its information content. Digital image processing involves the use of a digital computer for the required operations. In the case of images transmitted from spacecraft, the images are received at ground stations in the form of a stream of binary-coded data bits which are recorded on magnetic tape. The data can be converted to pictures by means of a straightforward process involving a film recorder. However, the results are often unsatisfactory in connection with geometric, photometric, and other types of distortion. The elimination of distortion by means of data manipulations conducted with the aid of computers is discussed, taking into account photometric manipulation, geometric correction, precision registration, and image enhancement [16]. According to Rafael C. Gonzalez [17], an image may be defined as a two-dimensional

function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of at any pair of coordinates (x, y) is called the Intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements and pixels. Pixel is the term most widely used to denote the elements of a digital image. Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception.

2.2.1 Coordinate Convention

The result of sampling and quantization is a matrix of real numbers. Let us assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size MN*. The values of the coordinates are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at (x, y) = (0, 0). The next coordinate values along the first row of the image are (x, y)= (0,1). The notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Figure 2.1 shows this coordinate convention. In the figure x ranges from 0 to M-1 and y from 0 to N-1 in integer increments. The coordinate convention used in the Image Processing Toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (x, y), the toolbox uses the notation (r, c)to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate tuple, (in the figure) refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M, and c from 1 to N, in integer increments. The result of sampling and quantization is a matrix of real numbers. We use two principal ways in this book to represent digital images. Assume that an image f(x, y) is sampled so that the resulting image has M rows and N columns. We say that the image is of size MN*. The values of the coordinates are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at (x, y) = (0, 0).

The next coordinate values along the first row of the image are (x, y)= (0, 1). The notation (0, 1) is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. [17]

Figure 2.1 shows this coordinate convention. Note that x ranges from 0 to M-1 and y from 0 to N1 in integer increments. The coordinate convention used in the Image Processing Toolbox to denote arrays which are different from the preceding paragraph in two minor ways. First, instead of using (x, y), the toolbox uses the notation (r, c) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate tuple (a, b), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at (r, c) = (1, 1); thus, r ranges from 1 to M, and c from 1 to N, in integer increments. Figure 2.1(b) illustrates this coordinate convention. Image Processing Toolbox documentation refers to the coordinates in Fig. 2.1(b) as pixel coordinates [17]

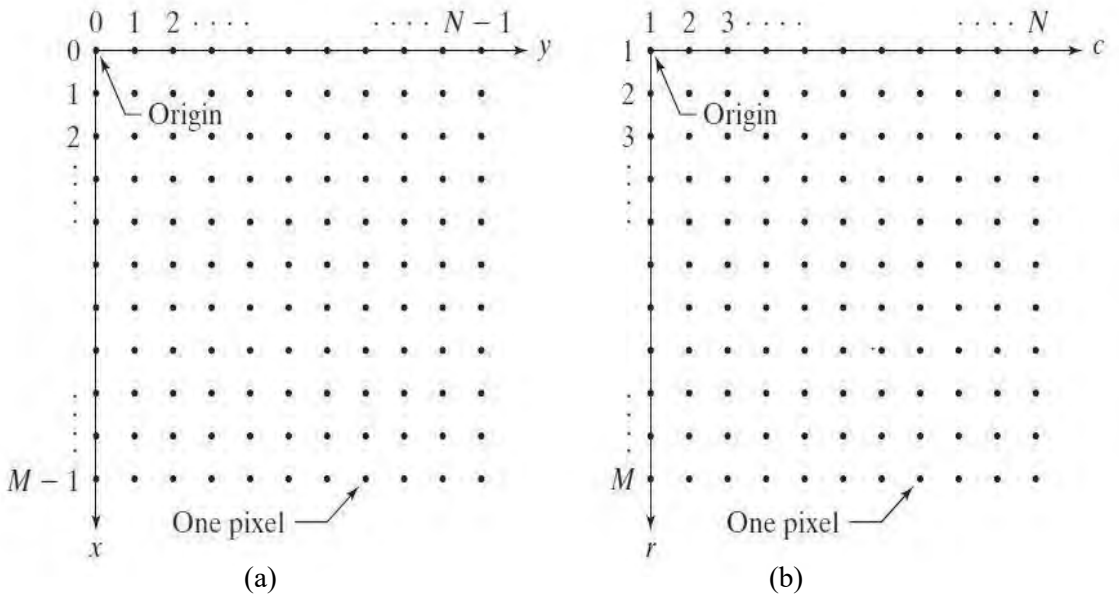


Figure 2.1 (a) Coordinate Convention (n-1) & 2.1(b) Coordinate Convention (n)

2.2.2 Matrix form of images

The coordinate system in Fig. 2.1(a) and the preceding discussion lead to the following representation for a digitized image:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, and pixel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements. A digital image can be represented as a MATLAB matrix:

$$\mathbf{f} = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(M,1) & f(M,2) & \cdots & f(M,N) \end{bmatrix}$$

Above we used the letters M and N, respectively, to denote the number of rows and columns in a matrix. A 1 X N matrix is called a *row vector*, whereas an M X 1 matrix is called a column vector. A 1 X 1 matrix is a scalar.

2.3 Image Segmentation

Image segmentation is to express the image as a physically meaningful connected region. The image segmentation purpose is often achieved through the analysis of different features of images such as edge, texture, color, brightness and so on. Image segmentation is usually to further the image analysis, identification, tracking, understanding, compression coding and so on, the segmentation accuracy directly affects the effectiveness of follow-up task, thus has a great significance. Image segmentation is the technology and process to divide the image into regions with different characteristics and extract the interested objectives [19].Image segmentation is to distinguish the different regions with special meaning based on image intensity, color, or geometric properties, and these regions are disjoint, every region meets the specific consistency [18].

2.3.1 Morphological Image Processing

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels

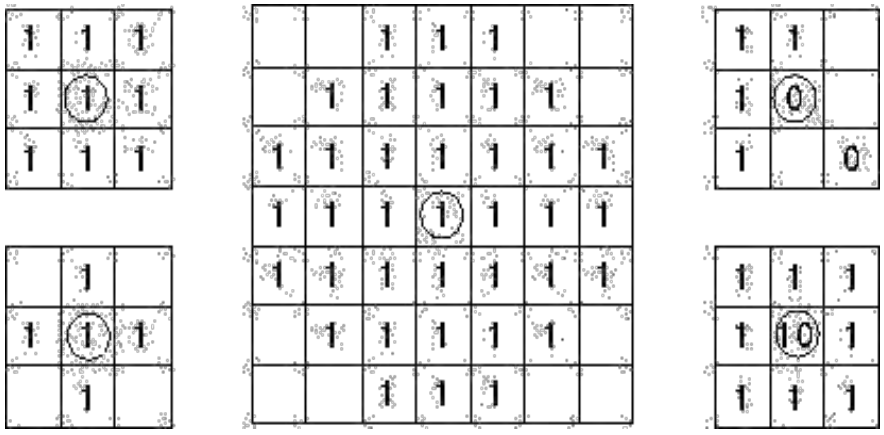
defines the operation as a dilation or an erosion. This table lists the rules for both dilation and erosion.

Rules for Dilation and Erosion

Operation	Rule
Dilation	The value of the output pixel is the *maximum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.
Erosion	The value of the output pixel is the *minimum* value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.

2.3.1.2 Structuring Elements

The structuring element is sometimes called the *kernel*, but we reserve that term for the similar objects used in convolutions. The structuring element consists of a pattern specified as the coordinates of a number of discrete points relative to some origin. Normally Cartesian coordinates are used and so a convenient way of representing the element is as a small image on a rectangular grid. Figure 1 shows a number of different structuring elements of various sizes. In each case the origin is marked by a ring around that point. The origin does not have to be in the center of the structuring element, but often it is. As suggested by the figure, structuring elements that fit into a 3×3 grid with its origin at the center are the most commonly seen type.



Note that each point in the structuring element may have a value. In the simplest structuring elements used with binary images for operations such as erosion, the elements only have one value, conveniently represented as a one. More complicated elements, such as those used with thinning or grayscale morphological operations, may have other pixel values.

An important point to note is that although a rectangular grid is used to represent the structuring element, not every point in that grid is part of the structuring element in general. Hence the elements shown in Figure 1 contain some blanks. In many texts, these blanks are represented as zeros, but this can be confusing and so we avoid it here.

When a morphological operation is carried out, the origin of the structuring element is typically translated to each pixel position in the image in turn, and then the points within the translated structuring element are compared with the underlying image pixel values. The details of this comparison, and the effect of the outcome depend on which morphological operator is being used.

2.3.1.3 Opening

The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels. However it is less destructive than erosion in general. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve foreground regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels.

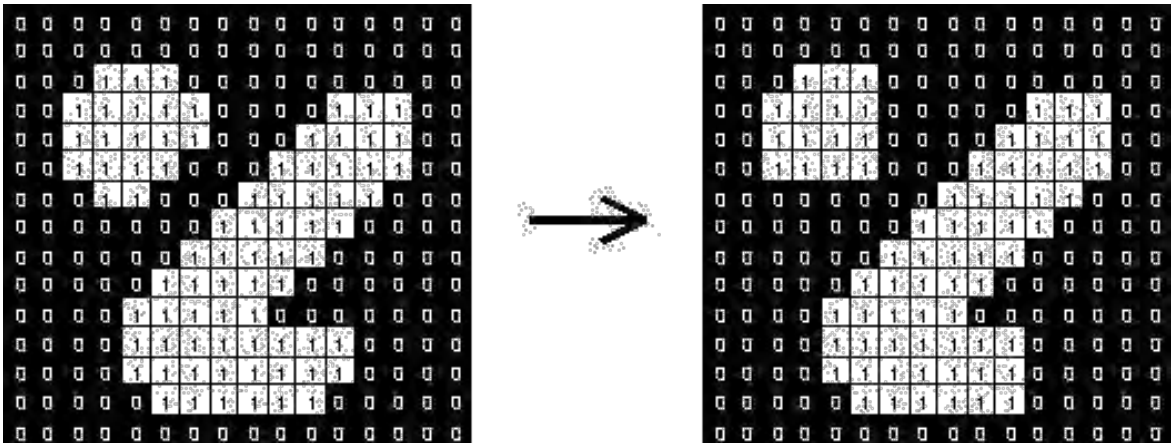


Figure 3.3.1.3: Effect of "Opening" using a 3 X 3 square structuring element

2.3.1.4 Closing

Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

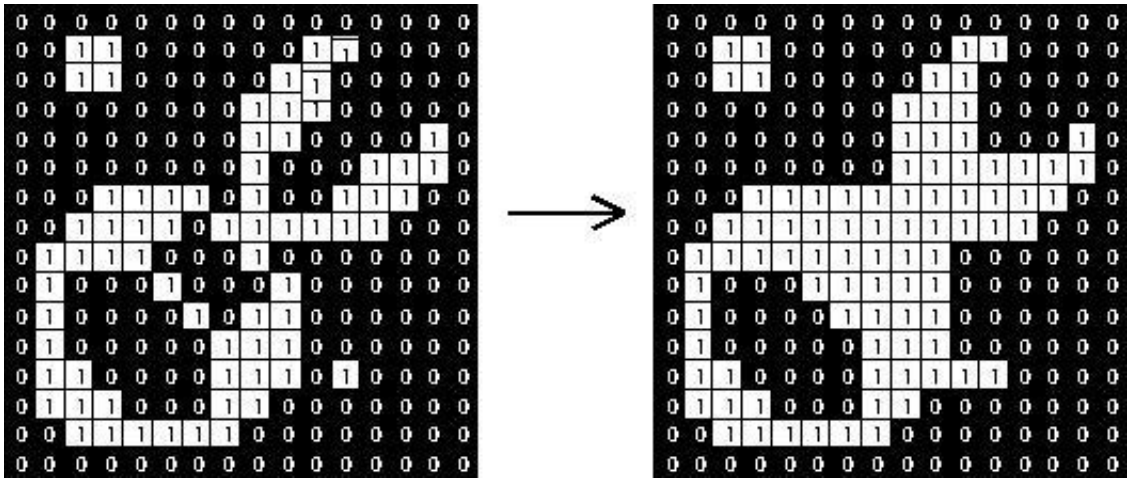


Figure 2.3.1.4: Effect of "Closing" using a 3×3 square structuring element

2.4 Color Conversion

According to Wan Najwa [19], “threshold” is an image segmentation to convert gray-scale to binary image. During the threshold process, individual pixels in an image are marked as “object” pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as “background” pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of “1” while a background pixel is given a value of “0.” Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's label.

2.5 Background subtraction

Background subtraction, also known as Foreground Detection, is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing (object recognition etc.). Generally an image's regions of interest are objects (humans, cars, text etc.) in its foreground. After the stage of image preprocessing (which may include removing image noising, post processing like morphology etc.) object localization is required which may make use of this technique. Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called “background image”, or “background model”. Background subtraction is mostly done if the image in question is a part of a video stream. Background subtraction provides important cues for numerous applications in computer vision, for example surveillance tracking or human poses estimation. However, background subtraction is generally based on a static background hypothesis which is often not applicable in real environments. With indoor scenes, reflections or animated images on screens lead to background changes. In a same way, due to wind, rain or illumination changes brought by weather, static backgrounds methods have difficulties with outdoor scenes [20].

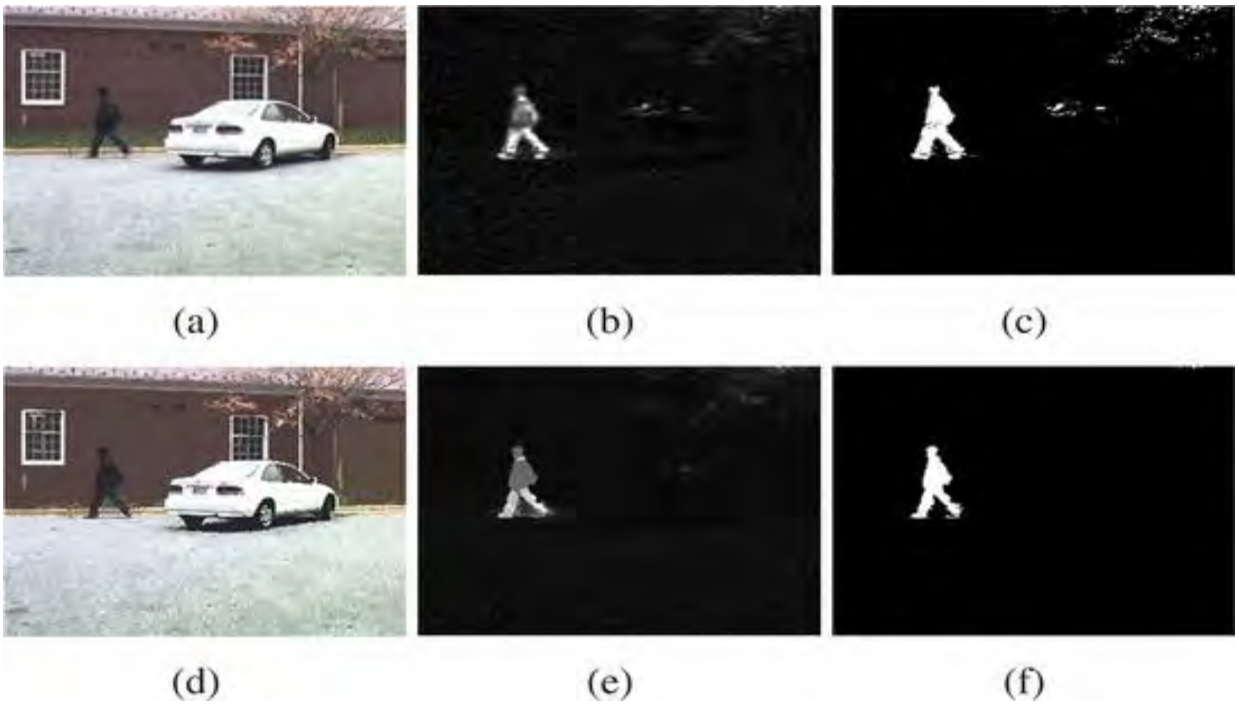


Figure 2.5 Background subtraction (a-f)

2.6 Using frame differencing

A motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take an image as background and take the frames obtained at the time t, denoted by I(t) to compare with the background image denoted by B. Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in I(t), take the pixel value denoted by P[I(t)] and subtract it with the corresponding pixels at the same position on the background image denoted as P[B].

In mathematical equation, it is written as:

$$P[F(t)] = P[I(t)] - P[B]$$

The background is assumed to be the frame at time *t*. This difference image would only show some intensity for the pixel locations which have changed in the two frames. Though we have seemingly removed the background, this approach will only work for cases where all foreground pixels are moving and all background pixels are static[20,21]. A threshold "Threshold" is put on this difference image to improve the subtraction

$$|P[F(t)] - P[F(t+1)]| > \text{Threshold}$$

This means that the difference image's pixels' intensities are 'threshold' or filtered on the basis of value of Threshold [22]. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds.

2.7 Mean filter

For calculating the image containing only the background, a series of preceding images are averaged. For calculating the background image at the instant *t*,

$$B(x,y) = \frac{1}{N}\sum_{i=1}^{N} V(x,y,t-i)$$

Where *N* is the number of preceding images taken for averaging. This averaging refers to averaging corresponding pixels in the given images. *N* would depend on the video speed (number of images per second in the video) and the amount of movement in the video [23].After calculating the background *B(x,y,t)* we can then subtract it from the image *V(x,y,t)* at time *t*=t and threshold it. Thus the foreground is

$$|V(x,y,t) - B(x,y)| > \text{Th}$$

Where *Th* is threshold. Similarly we can also use median instead of mean in the above calculation of $B(x,y,t)$.

Usage of global and time-independent Thresholds (same Th value for all pixels in the image) may limit the accuracy of the above two approaches [20].

2.8 Running Gaussian average

For this method, Wren [24] the author proposed fitting a Gaussian probabilistic density function on the most recent n frames. In order to avoid the fitting of from scratch at each new frame time t, a running (or on-line cumulative) average is computed.

The value of every pixel is characterized by mean $\mu_t$and varianceσ_t^2. The following is a possible initial condition (assuming that initially every pixel is background):

$$\mu_0 = I_0$$

$$\sigma_0^2 = <\text{Some default value}>$$

Here $I_t$is the value of the pixel's intensity at time$t$. In order to initialize variance, we can, for example, use the variance in x and y from a small window around each pixel.
Note that background may change over time (e.g. due to illumination changes or non-static background objects). To accommodate for that change, at every frame$t$, every pixel's mean and variance must be updated, as follows:

$$\mu_t = \rho I_t + (1-\rho)\mu_{t-1}$$

$$\sigma_t^2 = d^2\rho + (1-\rho)\sigma_{t-1}^2$$

$$d = |(I_t - \mu_t)|$$

Here ρdetermines the size of the temporal window that is used to fit the (usually$\rho = 0.01$) and $d$is the Euclidean distance between the mean and the value of the pixel.

We can now classify a pixel as background if its current intensity lies within some confidence interval of its distribution's mean:

$$\frac{|(I_t - \mu_t)|}{\sigma_t} > k \longrightarrow Foreground$$

$$\frac{|(I_t - \mu_t)|}{\sigma_t} \leq k \longrightarrow Background$$

Where the parameter k is a free threshold (usually$k = 2.5$). A larger value for $k$allows for more dynamic background, while a smaller $k$increases the probability of a transition from background to foreground due to more subtle changes.

2.9 RELATED WORKS

Video surveillance has been a major topic for research in recent years. Many approaches have been undertaken in analyzing and classifying video events. For example, object tracking [3], pedestrian detection [4], crowd counting [5], background modeling [6], action detection [7] etc. Suspicious event detection can mainly be divided in different types.

The usual approach is to first define the normal behavior and its properties, and then classifying those behaviors which do not have similar properties as the predefined normal behavior as suspicious events, because not many examples of suspicious behavior are available, and also it is impossible to completely determine the types of suspicious events that might occur. So it makes more sense to determine the properties of normal events, which occur frequently. Some of the approaches [8] are based on a set of constraints that are introduced to specify the normality, whereas the methods [9] are unsupervised approaches which directly determine normal patterns. The approach by Adam et al. [10] can be deemed local since the attention is directed to individual activities occurring in a local area. While this approach provides good results when it comes to implementation and efficiency, its performance suffers from the incapability of the model to incorporate temporal aspects of relationships among activities. Zhong et al. [9] detect objects by

thresholding a motion filter and they propose an unsupervised method that integrates the prototypical image features and classifies a group of behavior patterns either as normal or suspicious. Kim and Grauman [11] proposed a method to detect suspiciousities in a video sequence based on a space-time Markov random field model. This model dynamically adapts to suspicious activities that consists of unpredictable variations.

CHAPTER 03

PROPOSED MODEL

3.1 Introduction

Figure 3.1 demonstrates implementation a detailed of our proposed model. It demonstrates how the algorithm is set up.

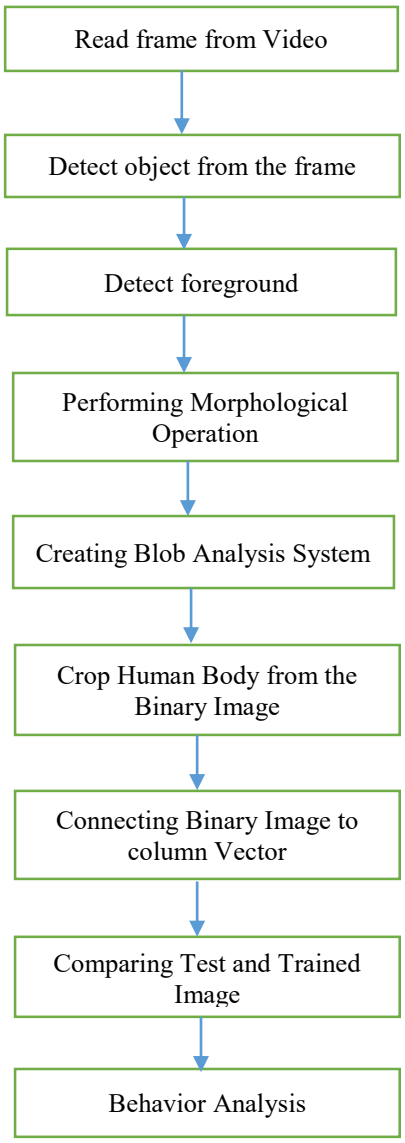


Fig 3.1 Proposed System Model

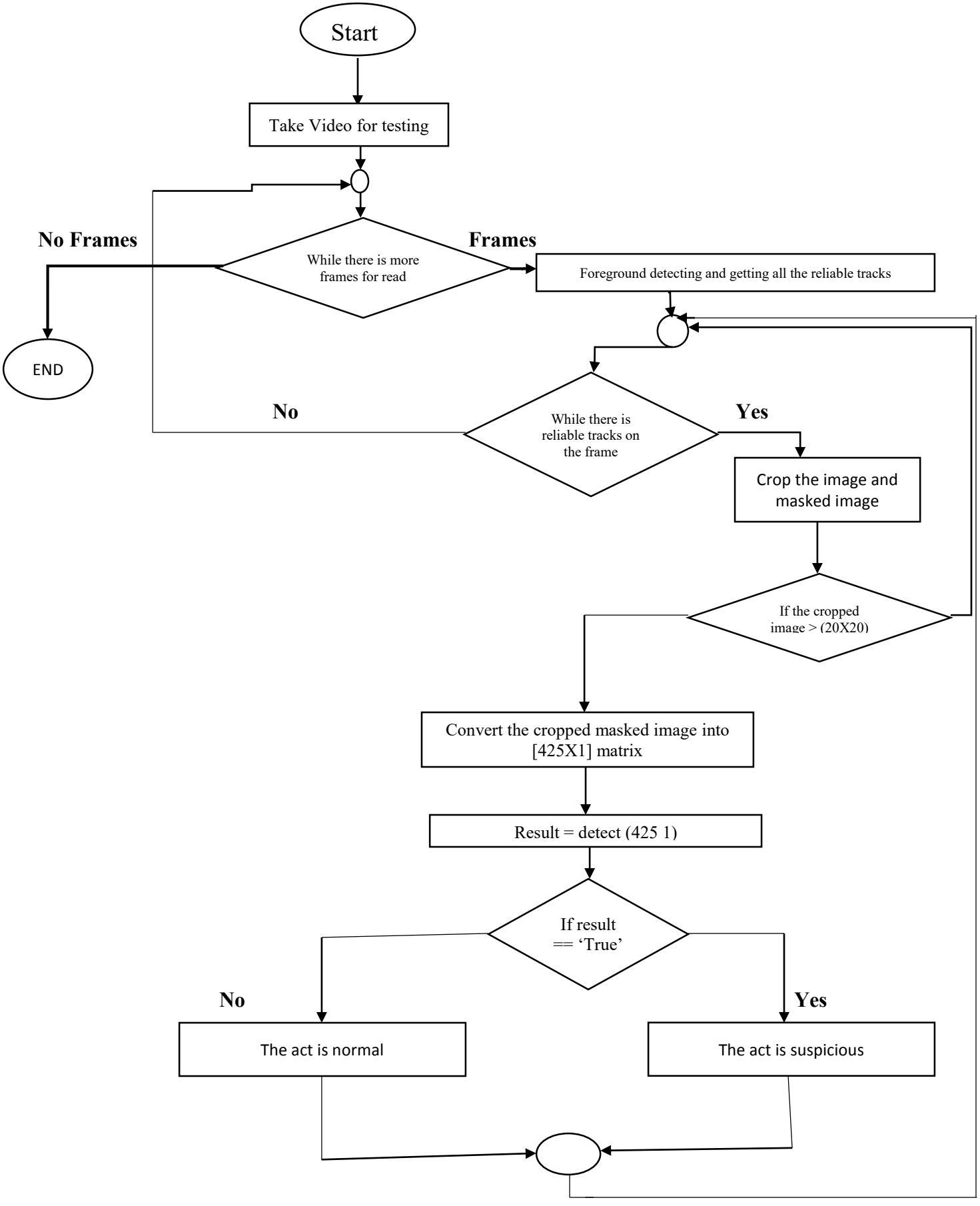


Fig 3.2: System Workflow

3.3 Create System Objects

Create System object is used to read frames from video, detecting foreground objects and displaying results. At first, we needed to initialize the video I/O. After that, we needed to create objects from reading video from a particular file. After that, we enabled to read every from video file while video was running. Since we wanted to see the video, we created two video players. First one was to display the original video while another one was to display the foreground mask. Then we needed to create system objects for foreground detection and blob analysis.

3.3.1 Create System Objects for Foreground Mask

The foreground detector is used to separate moving objects from the background. It returns a result which is a binary mask. Here the detector returns result based on Gaussian Mixture Model (GMM). In binary mask, the pixel value sets to be 1 (white) corresponding to the foreground. On the other hand, the pixel value sets to be 0 (Black) corresponding to the background.

In order to create Foreground detector we needed to pass some value as an argument. Here the number of Gaussian modes is 3 in the mixer model, number of training frame is 40 and minimum background ratio is set to be 0.7. This is the standard argument for the Foreground detector [6].

3.3.2Create System Objects for Blob Analysis

Blob Analysis is a fundamental technique of machine vision based on analysis of consistent image regions. As such it is a tool of choice for applications in which the objects being inspected are clearly discernible from the background. Diverse set of Blob Analysis methods allows to create tailored solutions for a wide range of visual inspection problems [26].
The blob analysis system object is used to draw a shape into the foreground detected objects. In order to draw the shape, it maintains a bunch of rules. First of all, it finds out in foreground objects how many pixels are gathered together. In other words, how many foreground pixels are connected with each other and form a group. Basically, the blob analysis system object is used to find out such groups. Foreground pixel contains the binary value of 1. So it identifies the group of value 1. These pixels are known as connected components. To find out these groups, the system objects emphasize on some characteristics such as area, centroid and the bounding box.

In order to create the blob analysis system objects, some values needed to be passed as arguments. Here Bounding Box Output Port is true, Area Output Port is true, Centroid Output Port is true and Minimum Blob Area is set to be 400.

3.4 Read Frame from Video

As we mentioned earlier, we needed to choose a frame manually, where the suspicious behavior exists. In order to it, we ran the video and picked up our desired frame. After that, we sent it to go through other operations. This is how, frames were chosen in our experiments.

Figure 3.4: Read Frame from Video

3.5 Detect Objects from Frame

The function returns the centroids and the bounding boxes of the detected objects. It also returns the binary mask, which has the same size as the input frame. Pixels with a value of 1 correspond to the foreground, and pixels with a value of 0 correspond to the background.

At first, the function performs motion segmentation using the foreground detector. After that it performs morphological operations on the resulting binary mask to remove noise from the pixel and finally fill the holes in the remaining blobs.

3.5.1 Detect foreground

The foreground detector System object compares a color or grayscale video frame to a background model to determine whether individual pixels are part of the background or the foreground. It then computes a foreground mask. By using background subtraction, we can detect foreground objects in an image taken from a stationary camera.[27]We created the Foreground detect system objects in upper step. Now using this object, we detected foreground objects. The System object compares a color or gray-scale video frame to a background model to determine whether individual pixels are part of the background or the foreground. It then computes a foreground mask which is also known as binary mask. The foreground detector requires a number of video frames and minimum background ratio in order to initialize the Gaussian Mixture Model (GMM). Here we used number of frames 40 and Minimum Background ratio 0.7. The idea of choosing these values is to provide us optimum result for every frame. The foreground segmentation process is not perfect and sometimes includes noise which makes the efficiency lower. In order to remove noise we have to do some operations which described below:

3.5.1.1 Morphologically Open Image

Morphologically open image function performs morphological opening operation in gray scale image or binary image with the structural elements that we created in above. The morphological open operation consists of two operations. At first, there is an erosion of an image and after that dilation operation is performed.

3.5.1.1.1 Creating Structural Element

Structural elements are used in structural analysis to split a complex structure into simple elements. Within a structure, an element cannot be broken down (decomposed) into parts of different kinds (e.g., beam or column) [28].Structural element is an essential part of erosion and dilation. It is mainly used to examine on input image. Two-dimensional, or flat, structuring elements consist of a matrix of 0's and 1's.The center pixel of the structuring element, called the origin. In our experiment, we create a structural element using which creates a flat, linear structuring element. Here Rectangle is indicated the shape and [3, 3] represents a 3 by 3 matrix which actually implements on neighbor's pixels. Using this element, we performed morphological operation like erosion and dilation.

3.5.1.1.2 Eroding Gray Scale Image

This operation erodes the binary image or gray scale image. The created structural element is passed here as an argument. After that the structural element executed over the gray scale image and returns the final eroded image. Here our corresponding image is a binary image, so it performs binary erosion. After erosion operation, the image lost thickness and details of the image.

3.5.1.1.3 Dilating Gray Scale Image

This operation dilates the binary or gray scale image. The created structural element is passed here as an argument. After that the structural element executes over the gray scale image and returns the final dilated image. Here our corresponding image is a binary image. So it performs binary dilation. It is used to thicken the objects in a proper way. Since after erosion operation, the image lost thickness and details, Dilation operation brings back the image closed to the previous one.

3.5.1.2 Morphologically Close Image

This function performs morphologically closing on the gray scale or binary image. At first, we needed to create the structural element. Bu using this elements it performs morphologically image closing. It is used to fill up the gaps of white pixels in binary images.

3.5.1.2.1 Creating Structural Element

The structural element is mainly used to examine on input image. Two-dimensional, or flat, structuring elements consist of a matrix of 0's and 1's.The center pixel of the structuring element, called the origin. In our experiment, we create a structural element using which creates a flat, linear structuring element. Here Rectangle is indicated the shape and [15, 15] represents a 15 by 15 matrix which actually implements on neighbor's pixels and provides the desired output.

3.5.1.3 Fill up The Holes in Binary Image

In this step, we used a function that fills up the holes in binary image. In upper method, we use morphologically close image to fill up the gap of white pixels but unfortunately, sometimes it fails to reach some white pixel. That is why, we used this method.

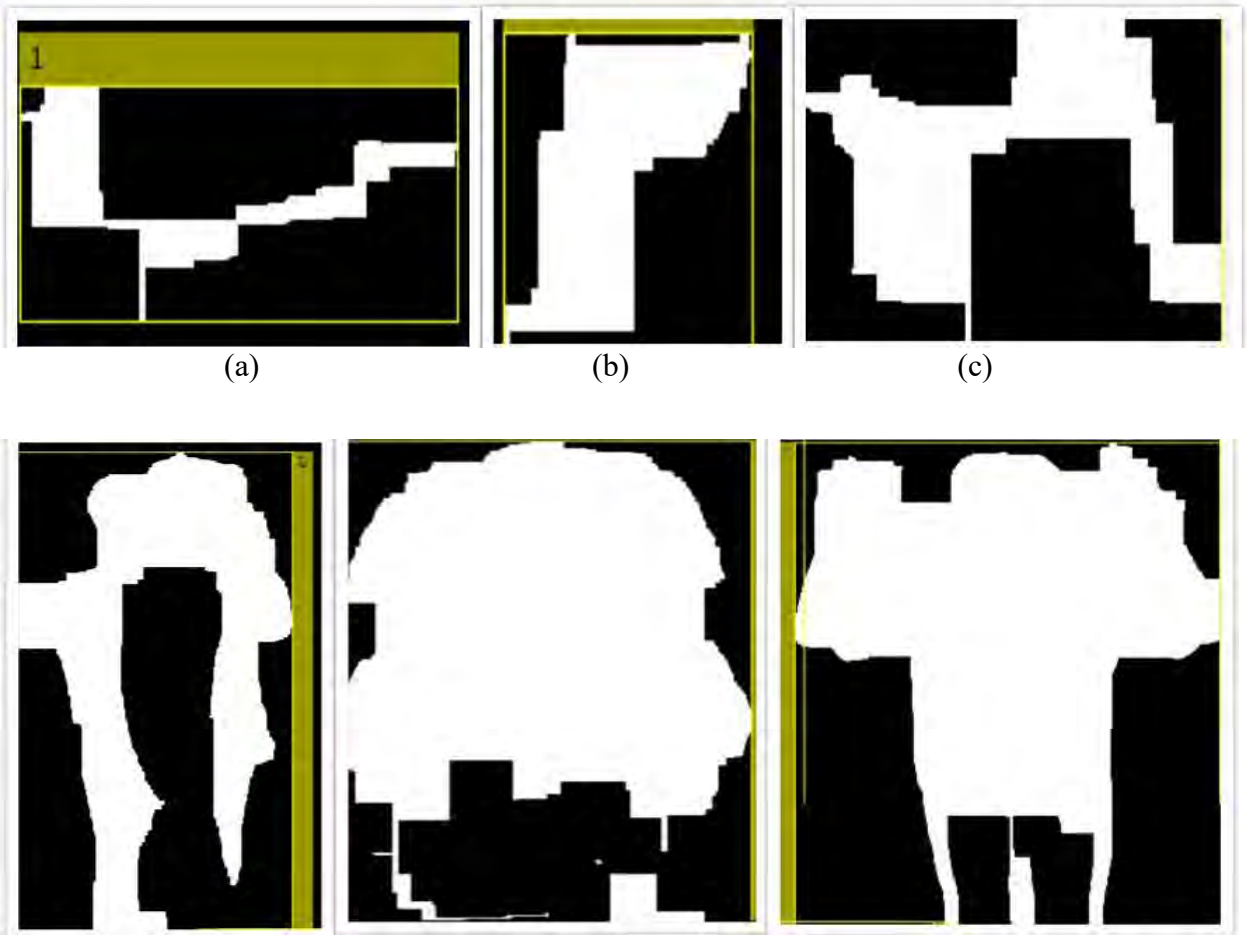
3.5.2 Blob Analysis

In above, we mentioned that we created a blob analysis system object to perform this operation. The operation is performed based on some characteristics like area, centroid and bounding box. As the functions finds out some groups of connected white pixels, we needed to give a shape on them. To do so, we first saved the result in a variable. After that we created a system object for shape inserter. So now, we used that objects into that variable where the detected human object was stored. The reason of selecting Rectangle shape is, we were interested only o human and rectangle shape was the best way to represent a human rather than using disk or square shape. Finally, we got a binary image where a human object was detected bounding by a rectangular box and stored in a variable.

3.6 Crop Human Body from the Binary Image

Now we got a binary image with human. We did not need the whole binary image. Here we observed whether the behavior was suspicious or not. So we only needed the human portion. We saved the value of human portion in a variable. In order to get it, we cropped the value from the

binary image. The idea of cropping image is to increase processing time and eliminate the noise. The next part was to train the cropped binary image.



(a) (b) (c)

(d) (e) (f)

Figure 3.6 Binary Image Conversion (a-f)

3.7 Training

This part is the most important part of our experiment. The efficiency of the output depends on how efficiently the data is trained. This training part consists of some steps. Among them, the first step is pre-processing.

3.7.1 Pre-Processing

In the preprocessing phase every binary image has to be resized into 425 X 125 pixels. Reducing the image size to 425 X125 decreases processing time and space.

3.7.2 Converting Binary Image to Column Vector

From above we acquired the binary image which has size of (200, 200). So we can consider the binary image as matrix having 200 rows and 200 columns. The matrix holds the binary value of 1 and 0. Precisely, we can say that the matrix stores binary value of 1 where the part of human body is found. Otherwise, it stores 0. Now we have to compute the value by iterating every column of each row. It means every row has 200 columns containing binary value. As the value is binary, we need to convert them into decimal value. In order to do so, we use the formula of binary to decimal conversion given below:

Row1=2^124*val +2^123*val+……..+ 2^1*val+ 2^0*val
|
|
|
Row200=2^124*val +2^123*val+……..+ 2^1*val+ 2^0*val

Now we need to store the value. To do so, we created a matrix having size of (200, 1) which has exactly same rows of the binary image. Actually, this is the column vector which is store the decimal value of each row. After calculating row1, the value was stored at column vector of (1, 1). After calculating row2, the decimal value was stored in column vector of (2, 1) and so on. This is how the column vector stored the decimal value of a binary image. Since we trained 50 images, we got 50 column vectors. After that we saved all column vectors in a global array. After training, we saved them in the computer hard drive. This value is needed when compared them with the testing column vector.

3.8 Processing with Testing Image

In our experiments, we are going to detect suspicious behavior from the video. So our input file is video which is full of suspicious behavior with different poses. As we did not find any universal data set, we made our own video as an input file. After reading a frame from input video, it made a decision if the frame contained suspicious behavior or not. It suspicious behavior exists, it gave

us warning. In other case, it just ignored the frame and checks the next frame. Processing the testing image consists of some steps.

3.9 Behavior Analysis:

In our dataset, the column vector of each binary image having size of (200, 1) have been stored. Now, the test image is also converted into the binary image. Then we compute the binary image into a column vector with same size of trained column vector (200, 1). Now we find out which trained image is very close enough to the test image. In order to so, we need to calculate the Euclidian distance between test column vector and trained column vector.

3.9.1Euclidian Distance

Euclidian distance is the ordinary distance between two points. It shows how far two points are located from each other. The Euclidian distance formula is given below:

The Euclidean distance between point p and q is the

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

In Matlab, we use the formula which given below:

D = sqrt (sum ((G - G1). ^ 2))

Here G is the column vector of test image and G1 is the column vector of train image. The explanation of Euclidian distance formula is to subtract two points by element wise and then

adding them up by squaring the subtracting the result. After adding all results, finally we perform square root over the result and the final result is our desired Euclidian distance.

The Euclidian distance is calculated by drawing two curves in graph. One curve for test column vector and another one is for trained column vector. Below a graph is shown

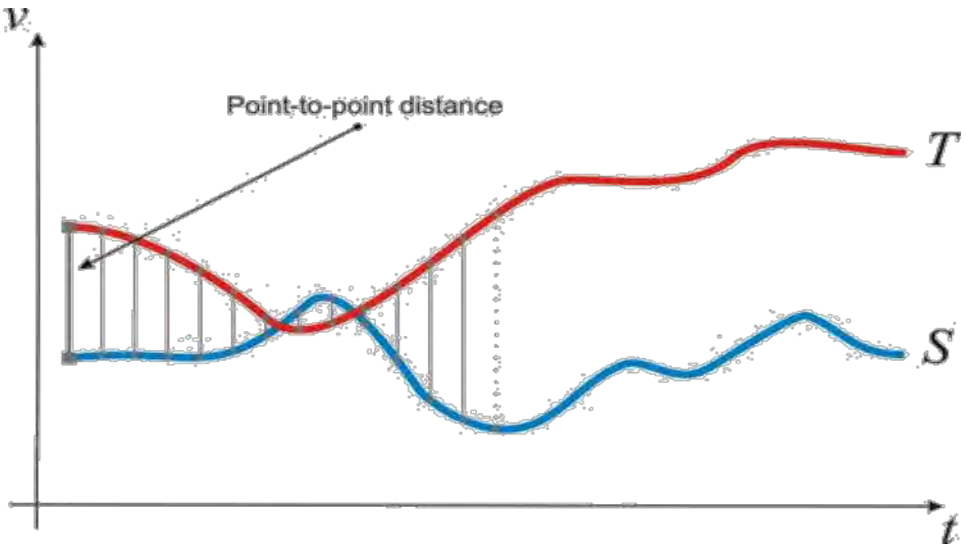


Figure 3.9.1: Euclidian Distance graph

T and S represent the curves of two column vectors. Between each pair of points, a line is drawn to indicate the point to point distance. After drawing the line, the Euclidian formula is used. This methodology will be followed for every pair of test and trained column vector. Then we store the result in an auxiliary array. The result shows that how two vectors are close to each other. It means it shows the similarities or between two vectors. In other words, we can say that the result tells us the difference between two column vectors. Now we find out the minimum value of an array. The minimum value indicates that the result of column vector is the best match or gets the highest similarity compared with test column vector. If the result is very close to 0 or exactly 0, then we can say that, the two column vectors are same. Hence, it is obvious that the test image and trained image is exactly same. As we only train the suspicious behavior of human, we can say that the test image contains suspicious behavior and applicable to draw further attention.

CHAPTER 4
EXPERIMENTAL RESULT & ANALYSIS

4.1 Setup

For our experiment, we used MATLAB R2017a and Image Processing Toolbox. Algorithm we used is Background subtraction Algorithm. Methodologies we followed are Morphologically Open Image, Subtracting Back ground from image, Converting Gray scale to Binary Image, Converting Binary Image to Column Vector etc.

4.2 Dataset

For this test, we could not locate any general picture set. In this manner we have to go extraordinary spots to gather pictures. The photos were taken by 6 mega pixel cell phone and 12 mega pixel advanced camera, and some of are gathered from web. As we are only interested in suspicious behaviors, so we took lots of images with different poses and same number of ratio. The pictures were taken in this way so that camera angle varied from human about 45 to 90 degrees. The distance between camera and human is about 3 to 5 meters.5 meters is a standard distance because increasing the distance can make the system inefficient. Less than 3 meters brings human too close. All the images are considered as training image. We have trained 300 images for this experiment.

Figure 4.2: Sample data collection for training purpose

4.3 Result

In our experiments, input file is video which is full of suspicious behavior with different poses. For detecting suspicious behavior from the video we made our dataset & trained them. After reading a frame from input video, it made a decision if the frame contained suspicious behavior or not. It suspicious behavior exists, it gave us warning. In other case, it just ignored the frame and checks the next frame. In our dataset, the column vector of each binary image having size of (200, 1) have been stored. Now, the test image is also converted into the binary image. Then we compute the binary image into a column vector with same size of trained column vector (200, 1). Now we find out which trained image is very close enough to the test image. In order to so, we need to calculate the Euclidian distance between test column vector and trained column vector.

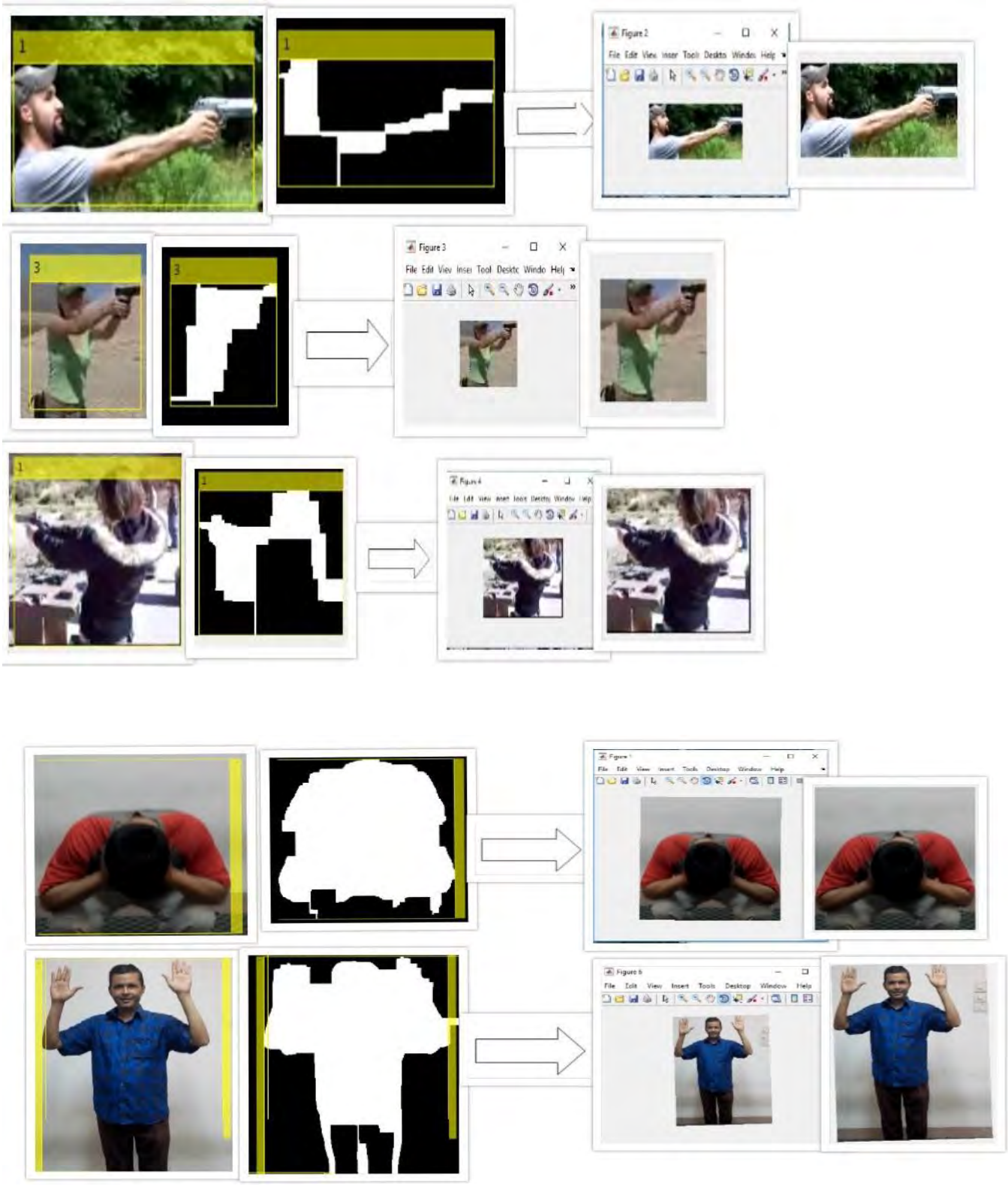


Figure 4.3 Test frame and Output

Table 1: Test image matched with trained image

Test Frames Based on Foreground Detection		
Frame Type	Test Frame	Correct Output
Frames with Normal Behavior	20	15
Frames with Suspicious Behavior	80	60
Total	100	75

Accuracy for Suspicious behavior using Foreground Detection Algorithm:

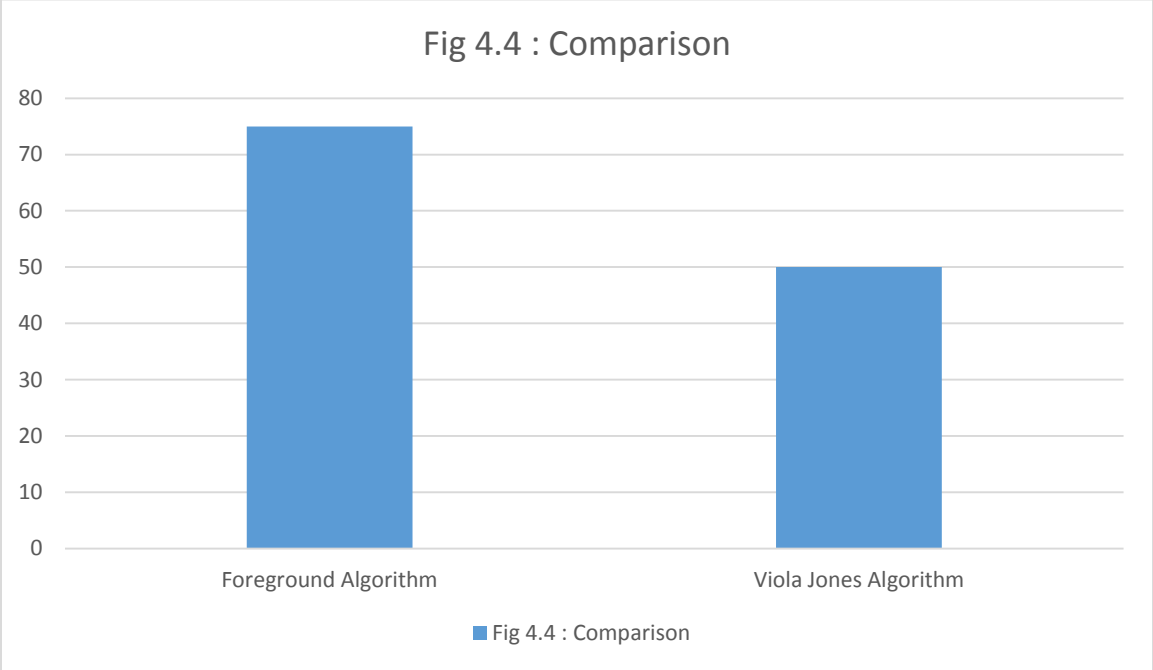
$$\frac{\text{Correct Output}}{\text{Test frame}} \times 100\%$$

$$= \frac{60}{80} \text{ x } 100\%$$

$$= 75\%$$

4.4 Comparison

In our system we first use viola jones algorithm to detect human body. However the problem is viola jones algorithm need front facial feature to detect. In many cases it is very hard to detect the face because of environment or for human's position. We find that the accuracy level is decreasing. In our experiments, the system using face detection (Viola Jones) algorithm achieved 50% accuracy whereas the system using foreground detection achieved 75% accuracy.



CHAPTER 05

CONCLUSIONS AND FUTURE WORKS

5.1 Conclusion

In this paper, we proposed an automated video surveillance system which can analyze the behaviors of human and detect the suspicious behaviors. In “chapter 1” we applied Face Detection method for detection purpose. Nevertheless, the accuracy we got by applying this approach was much less. Therefore, we moved to another approach which provided us much better result than the previous approach. The approach is „Foreground Detection Approach“. This approach uses GMM to extract the human portion from the background and later on using morphological image processing we converted the RGB image into binary image. Our proposed algorithm gave efficient results and a decent accuracy.

5.2 Future Work

In our next step of research, we plan to detect suspicious behaviors in real time. We are planning to create a connection between the surveillance system and a device which will be controlled by the operator in charge of the system. When our system will find any suspicious behavior it will immediately send the detected steaming part to the connected device. Thus the operator or the user who is in charge of that will be notified instantly. As a result, the operator’s do not have to monitor the surveillance videos all the time. By using this kind of intelligence system the necessity of manpower can be reduced.

REFERENCE

[1] M. Shah, O. Javed, and K. Shaque. Automated visual surveillance in realistic scenarios.IEEE Computer Society, 14:30-39, Jan.-March 2007.

[2] V. Gouaillier and A. Fleurant. Intelligent video surveillance: Promises and challenges.Technological and Commercial intelligence Report, March 2009.

[3] Avidan, S. (2005). Ensemble Tracking. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 261-271.

[4] Cong, Y., Gong, H., Zhu, S., & Tang, Y. (2009). Flow mosaicking: Real-time pedestrian counting without scene-specific learning. 2009 IEEE Conference on Computer Vision and Pattern Recognition.

[5] Dalal, N., &Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 886-893.

[6] Stauffer, C., &Grimson, W. (1999). Adaptive background mixture models for real-time tracking. Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149).

[7] 18 Yuan, J., Liu, Z., & Wu, Y. (2009). Discriminative subvolume search for efficient action detection. 2009 IEEE Conference on Computer Vision and Pattern Recognition.

[8] H. Dee and D. Hogg. Detecting inexplicable behavior. In BMVC, 2004

[9] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In CVPR, pages 819–826, 2004.

[10] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixedlocation monitors. PAMI, 30, 2008.

[11] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting suspicious activities with incremental updates. In CVPR, 2009.

[12] Suspiciousity (Behavior).” *Wikipedia*, Wikimedia Foundation, 3 Aug. 2017, en.wikipedia.org/wiki/Suspiciousity_(behavior)

[13] R. Mehran, A. Oyama, and M. Shah, “Suspicious crowd behavior detection using social force model,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

[14] M. D. Breitenstein, H. Grabner, and L. V. Gool, “Hunting Nessie – realtime suspiciousity detection from webcams,” in IEEE International Workshop on Visual Surveillance, 2009.

[15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In CVPR, 2010.
[16] Structuring Elements." *Glossary - Structuring Elements*, homepages.inf.ed.ac.uk/rbf/HIPR2/strctel.htm.

[17] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins "Digital Image Processing Using MATLAB® (Second Edition)" A Division of Gatesmark, LLC, 2009.

[18] Kim,Eun Vi, Jung Keechul, Genetic algorithms for video segmentation U].Pattern Recognition, January, 2005, 59-73. [19] Wan NajwaBinti Wan Ismail, Faculty of Electrical & Electronics Engineering University Malaysia Pahang, MAY, 2009.

[19] L.Ming. Image Segmentation Algorithm Research and Improvement. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE),2010

[20] B. Tamersoy (September 29, 2009). "Background Subtraction – Lecture Notes" (PDF). University of Texas at Austin.

[21]B. Patel; N. Patel (March 2012). Motion Detection based on multi-frame video under surveillance systems. Vol. 12.

[22]N. Lu; J. Wang; Q. Wu; L. Yang (February 2012). An improved Motion Detection method for real time Surveillance.

[23]Y. Benezeth; B. Emile; H. Laurent; C. Rosenberger (December 2008). Review and evaluation of commonly-implemented background subtraction algorithms (PDF). International Conference on Pattern Recognition. pp. 1–4. doi:10.1109/ICPR.2008.4760998.

[24]C.Wren; A. Azarbayejani; T. Darrell; A. Pentland (July 1997). "Pfinder: real-time tracking of the human body" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 19 (7): 780â€“785. doi:10.1109/34.598236. Archived from the original (PDF) on 2007-06-09.

[25]Stauffer, C. and Grimson, W.E.L,Adaptive Background Mixture Models for Real-Time Tracking, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol. 2 (06 August 1999), pp. 2246-252 Vol. 2.

[26]docs.adaptive-vision.com/4.7/studio/machine_vision_guide/BlobAnalysis.html+code+link+)https://www.mathworks.com/help/vision/examples/motion-based-multiple-object-tracking.html

[27]Stauffer, C. and Grimson, W.E.L,Adaptive Background Mixture Models for Real-Time Tracking, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, Vol. 2 (06 August 1999), pp. 2246-252 Vol. 2.

[28] Waddelln Alexander Low Waddell (1916). Bridge Engineering - Volume 2. New York: John Wiley & Sons, Inc. p. 1958. Retrieved 2008-08-19.