

Real Time Camera based Human Fall Detection Using Raspberry Pi

by

Bathula Sharath Kumar

A thesis submitted in partial fulfillment of the requirements for the
degree of Master of Engineering in
Microelectronics and Embedded systems

Examination Committee: Dr. Mongkol Ekpanyapong (Chairperson)
Dr. A.M. Harsha S. Abeykoon
Assoc. Prof. Erik L.J. Bohez

Nationality: Indian
Previous Degree: Bachelor of Technology in
Electronics and Communications Engineering
Jawaharlal Nehru Technological University, Hyderabad
Telangana, India

Scholarship Donor: AIT Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2019

ACKNOWLEDGEMENTS

I am grateful to everyone who have supported and helped me in the completion of this Master thesis.

Firstly, I would thank my advisor Dr. Mongkol Ekpanyapong of the Microelectronics and Embedded Systems (MES) in AIT (Asian Institute of Technology), for his continuous guidance, support and encouragement for this thesis study, his immense knowledge which helped me a lot in progressing in my thesis. I deeply appreciate the time and effort he kept on me.

I would also like to thank my committee members Dr. A.M. Harsha S. Abeykoon and Assoc. Prof. Erik L.J. Bohez for their valuable comments and suggestions which helped me improve my work.

Secondly, I extend my gratitude to the Royal Thai Government (RTG) and Asian Institute of Technology (AIT) fellowships for the AIT fellowship scholarship.

Finally I express my gratitude to my parents and friends for their continuous encouragement and unfailing support throughout the course and through the process of writing thesis. I specially thank my brother for his immense support and encouragement throughout this Masters, Thank you. This attainment of my work would not have been possible without them.

ABSTRACT

In today's elderly population, fall is the major health complication problem. Every year, around the world almost 30 percentage of falls occurs in elderly people aged more than 65. When fall was occurred most of them need assistance to stand up promptly. The longer they stays down on the floor after the fall the injuries becomes more severe as the time passes, so they should be assisted as soon as possible. A fall detection algorithm based on image processing techniques is so much useful in this situation.

This thesis makes use of OpenCV (Open Source Computer Vision) library for all the image processing techniques in python language. This thesis develops a fall detection algorithm and makes it possible in real time on Raspberry pi. In This system fall will be detected based on parameters vertical velocity, aspect ratio, angle, traveled distance and change in height of the elderly people and a mail will be sent through using SMTP protocol. Initially the background will be obtained when the subject is not in frame and after that subject will be tracked frame to frame, this tracking is useful in obtaining above parameters with respect to last frame location. Fall will be confirmed when the change in above parameters is abnormal. The partial objective human detection will be done using Histogram of Oriented Gradients and Linear SVM method. Finally using SMTP protocol we will sends an email whenever human fall is occurred. From the test results, this systems gives an overall of precision, recall and total classification of accuracy is 93.75%, 88.23% and 90.65%.

Keywords: Fall detection, Support Vector Machine (SVM), Histogram of Oriented Gradients (HOG), Simple Mail Transfer Protocol (SMTP), Open source Computer Vision (OpenCV).

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	TITLE PAGE	i
	ACKNOWLEDGEMENTS	ii
	ABSTRACT	iii
	TABLE OF CONTENTS	iv
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
	LIST OF ABBREVIATION	viii
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Statement of the Problems	2
	1.3 Objectives	2
	1.4 Limitations and Scope	3
	1.4.1 Limitations	3
	1.4.2 Scope	3
	1.5 Thesis Outline	3
2	LITERATURE REVIEW	4
	2.1 Introduction	4
	2.2 Non-computer vision based Techniques	5
	2.3 Computer vision based methods	7
	2.3.1 Analytical methods	7
	2.3.2 Machine learning algorithm	9
	2.4 Literature Survey Matrix	11
	2.5 Summary	12
3	METHODOLOGY	14
	3.1 System Overview	14
	3.2 Flowchart	15
	3.3 Hardware specifications	16
	3.4 Software setup	18
	3.5 Image Processing Techniques	18
	3.5.1 Video acquisition	18
	3.5.2 Background subtraction	18
	3.5.3 Object detection	19
	3.5.4 Morphology operations	20
	3.5.5 Tracking of the object	23
	3.6 Fall detection parameters	25
	3.6.1 Velocity of the object and travelled distance	25

3.6.2 Aspect Ratio and angle	26
3.6.3 Vertical Velocity	27
3.6.4 Height	28
3.6.5 Thresholds	28
3.7 Human detection	29
3.7.1 ROI selection	29
3.7.2 Human detection using Hog	30
3.7.3 Filtration of Detection	35
3.8 Simple Mail Transfer Protocol (SMTP)	36
4 EXPERIMENTAL RESULTS	38
4.1 Calculation of Accuracy	39
5 CONCLUSION AND RECOMMENDATIONS	41
5.1 Conclusion	41
5.2 Recommendations	41
REFERENCES	42

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1.1	Different types of Wearable fall detectors	1
Figure 2.1	Fall detection devices classification	4
Figure 3.1	Raspberry pi 3 interface with camera	14
Figure 3.2	Flow chart of system working	15
Figure 3.3	Hardware block Diagram	16
Figure 3.4	Interfacing camera module to raspberry pi	17
Figure 3.5	Enabling camera interface in raspberry pi	17
Figure 3.6	Background subtraction	19
Figure 3.7	Binary Image from the differenced image	20
Figure 3.8	Binary Image after erosion operation	21
Figure 3.9	Binary Image after dilation operation	22
Figure 3.10	Binary Image after closing operation	23
Figure 3.11	Coordination system of digital image processing	24
Figure 3.12	Tracking point	25
Figure 3.13	Standing state detection with AR and angle	26
Figure 3.14	Sitting state detection with AR and angle	27
Figure 3.15	Fall state detection with AR and angle	27
Figure 3.16	ROI extraction from input video	29
Figure 3.17	Human detection with HOG and SVM	30
Figure 3.18	Orientation Binning	31
Figure 3.19	Histogram of Gradients (8x8 cell)	32
Figure 3.20	Cells and blocks structure of HOG	32
Figure 3.21	Down sampling of image	33
Figure 3.22	Detector Windows	34
Figure 3.23	Filtrations of detections results	35
Figure 3.24	Elimination of overlapping rectangles	36
Figure 3.25	Source code to send email using SMTP protocol	37
Figure 4.1	Raspberry pi monitor display of fall detection	38
Figure 4.2	The Alert email sent to recipient when fall detected	39

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1	Comparison of non-computer sensor and computer vision based methods	11
Table 2.2	Literature Survey	12
Table 4.1	Experimental results obtained by fall detection	39

LIST OF ABBREVIATIONS

WHO	World Health Organization
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine
SMTP	Simple Mail Transfer Protocol
Rpi	Raspberry pi
CV	Computer Vision
MFCC	Mel Frequency Cepstral Coefficient
IP	Internet Protocol
T	Threshold
ROI	Region of Interest

CHAPTER 1 INTRODUCTION

1.1 Background

The falling is the major health complication problem especially for older people. In elderly people falling is a typical action in their everyday life. Sometimes this falls becomes more severe that, if the victim is not assisted with in time it leads to death. According to a survey conducted by the World Health Organization (WHO) estimates that 28% to 35% of elderly people aged more than 65 years goes through falls at least once a year [1], and it increases up to 44% for elderly people aged more than 70 years. Based on the surveys of the World Health Organization, falling hospitalizes more than 50% of the elderly people and falls are reasonable for almost 40% of all injury deaths [1].

Fall is especially complicated for the elderly people that living individually because a lot of time will be taken before they get assistance after the fall. According to World Health Organization (WHO) the elderly people population will be raised significantly over the upcoming 20 years.

Human falling can be detected mainly based on three approaches this includes wearable devices, non-wearable (ambience) sensor based devices and finally the vision based devices. The wearable and non-wearable devices are made of tiny electronic sensors. The ambience based devices consists of sensors they do not need to wear. This wearable and non-wearable devices such as belts, pressure sensors and floor vibration sensors are easy to use and the cost is minimal but this kind of devices are not reliable they prone to false alarms. (See Figure 1.1) shows different types of wearable detectors wearing at different parts of the body to detect fall. Elderly people especially who have dementia, often forgets to wear and charge these devices. Due to these false warnings, poor reliability, battery conditions and wearing problems users and health care centers won't recommend wearable and non-wearable fall detectors.



Figure 1.1: Different types of Wearable fall detectors. Reprinted from the Research paper [2]

The human fall detection system that we proposed in this thesis is a low-cost vision based device, it's consists an embedded computer (Raspberry Pi) and camera. This system can be setup on walls or ceilings, it will starts detection without any human interference. In this system the elderly people are not required to wear any devices for detection. This system is particularly designed for elderly people that are alone because, if any other persons are around there and fall happens then they will help each other respectively.

1.2 Statement of the Problems

The wearable and non-wearable fall detectors are not reliable due to their higher false alarm rates, battery requirements, too sensitive, forgetting of wearing and needs higher maintenance. Nowadays, in most of cases, vision based fall detectors are used for fall detection. Current vision based fall detectors takes subject postures as their main parameter for fall detection due to this these detectors cannot differentiate between accidental fall and sleeping/lying on floor. They don't use any human detection techniques so they will consider both human falls and other objects falls as same and generates false alarm. These are not productive, they all have to compute complex image processing techniques and they have to run fast to obtain a real time detection so they have to run on a standard computers. So they becomes so expensive and less portable.

1.3 Objectives

The main objectives of the proposed system are as follows:

- To develop a camera based human fall detection on a lower performance embedded computer device (Raspberry Pi).
- Developing foreground feature extraction from static background.
- Developing contours and extracting features from silhouette.
- Developing foreground feature extraction for tracking of the Elderly people.
- Obtaining the parameters such as vertical velocity, aspect ratio, traveled distance and change in height of the elderly people using different algorithms.
- Developing a robust algorithm for fall detection based on parameters vertical velocity, aspect ratio, traveled distance and change in height of the elderly people.
- To develop a robust human detection by applying a ROI scheme for real time human detection using Histogram of Oriented Gradients (HOG) in conjunction with State Vector Machines (SVM).
- Alerting through email using SMTP protocol.

1.4 Limitations and Scope

1.4.1 Limitations

- Cannot be used for detecting fast moving objectives due to Raspberry Pi in-capabilities.
- The whole detection will be taken in static environment or in a closed room.
- Reflective surfaces are challenging.

1.4.2 Scope

- Aims at increasing accuracy of detecting human's movements.
- Faster in human detection and computation.

1.5 Thesis Outline

I organize the rest of this dissertation as follows.

In Chapter 2, I describe the literature review.

In Chapter 3, I propose my methodology.

In Chapter 4, I present the experimental results.

Finally, in Chapter 5, I conclude my thesis.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

As stated in first Chapter, for elderly people especially that leaving individually at home needed efficient human fall detectors to detect the fall events and sent that information to respective care takers or hospitals for assistance. In this Literature review, different surveys on fall detection techniques are explained in an organized way as shown in below figure (see figure 2.1).

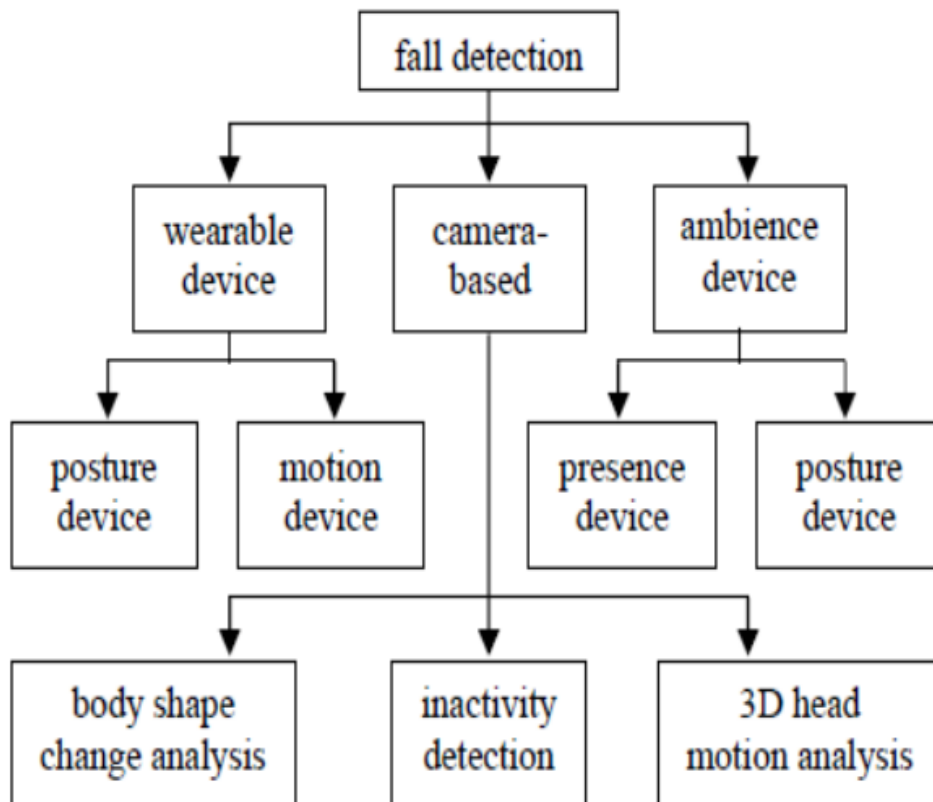


Figure 2.1: Fall detection devices classification [3]

Further this fall detection techniques can be divided into two main types:

1. Non-computer vision based (wearable and ambience)
2. Computer vision based (camera based)

2.2 Non-computer vision based Techniques

There are so many different types of non-computer vision based techniques are there for fall detection. In these techniques, various sensors (floor vibration sensors, acceleration sensors, acoustic sensors and pressure sensors) are used to obtain the vibration, sound and body motion parameters to detect the fall.

In this paper [4], it develops a real time system for human movement detection using a single tri-axial accelerometer mounted at waist of elderly people. This system is a tiny and wireless device and it consumes very less power for working. Acceleration signals will be generated based on gravity and using various digital filters body movement will be sampled and processed. The various filters used are median filter, high pass filter (HPF) seventh order and low pass filter (LPF) second order. A classifier was then developed and applied on the processed data for detection of fall and for the classification of various movements out the elderly people by second to second approach. The main majority and complex steps will be performed online in the embedded microcontroller, and when fall was occurred, for further analysis, data will be sent to a local computer (receiver in this paper).Based on experimental results the system stated in this paper gives an overall accuracy of 95.6% for the possible fall detection activities. Unfortunately, this system cannot distinguish between falls and sitting, because the generated acceleration signals were almost similar for falling and sitting actions.

In this paper [5], they used the same single tri-axial accelerometer as used in [4]. They just changed the position of the accelerometer sensor position from waist position to different positions of the subject's body such as wrist, head and waist for the fall detection activity. They measured the accelerometer readings from different positions by the accelerometer and they compared those measures with a proper threshold for the detection fall activity. This system is efficient than previous paper [4], the results proved that accelerometer used at wrist and head are more efficient than wearing that at waist. By using a simple thresholds algorithm this system stated in [5] gave a specificity of 1000% and sensitivity of 97-98%.

An acceleration sensor can be joined with other devices to obtain a more reliable and efficient fall detection system. In this paper [6], an accelerometer sensor was joined with other sensors to obtain an intelligent platform for the fall detection. Accelerometer sensor acted as the main sensor in it. In this paper, the architecture of the device is proposed in a multilayer passion. The first layer consists a unit of several acceleration sensors, this layer is an intelligent accelerometer unit (IAU). IAU obtains all the acceleration information. The second layer manages the communication with peripherals (like a multi-person intelligent platform (MIP), this is an accessing point to the tele-hospital center) this layer is called as personal intelligent platform (PIP). PIP access the acceleration data from the first layer and performs fall detection algorithm. For detection of the fall a double-threshold analysis is used, the angle threshold and magnitude threshold were compared to obtain the fall detection.

This paper [7] is an extension to the paper [6], in this paper both algorithm and hardware aspects are been improved. An accelerometer smart sensor (ACSS) is a small and low powered

waterproof biocompatible sensor. ACSS sensor is used and additionally in the second layer user interference module was introduced. Second layer is a personal server (PSE), it allows the elderly people to access some of the important information. A time analysis is additionally used in the fall algorithm, so certain waveforms of accelerometer information will be obtained with respect to time. This helps in some problematic fall detections such as knee falling. The system proposed in this paper [7] gives an overall sensitivity of 100% and specificity of 95.68%. We can further reduce the false positives manually accessing information using user interface.

In this paper [8], The Sensor of Movements (SoM) is an accelerometer based smart sensor. The SoM will performs the fall event detection or first energy detection. Whenever the fall event was detected, 4 seconds of accelerometer information around the event was sent as an activity pattern to ADM (Analysis and Decision) module to check for fall detection. ADM will uses the double-threshold analysis stated in paper [6] and [7]. The main contribution of this paper is an optimized module (OM). OM finds out the optimum parameters for the event detection with SoM and the fall detection with ADM. Personalization and adaption is possible with OM, it uses new obtained activity pattern readings to keep updating the threshold parameters or updating the process steps implemented in modules of SoM. This paper gives a 100% of success for event detection (due to optimization of thresholds by OM), sensitivity of 100% and specificity of 95.68%.

In non-computer vision methods beside accelerometer we will also use some other sensors too, such as vibration and acoustic sensors were also applied.

In this paper [9], this system is an acoustic based fall detection system, it detects falls automatically and reports it to the care takers. This system has built with a circular array (consists of 8-microphones), so it gives a good 3-D estimation of acoustic (sound) location, by utilizing the algorithm called SRP-PHAT (steered response power with phase transform) from paper [10], and also by using beam forming technique the sound signal will be enhanced. From this enhanced sound signal the MFCC (Mel frequency cepstral coefficient) feature will be extracted and to discriminate between fall and non-fall actions they uses the k-th nearest neighbor technique. For testing they took a dataset that contains 120 non-fall actions and 30 falls actions. It has detected all the falls activities correctly but 6 non-fall activities were also showed as fall activities. The paper [11] was an improvement of the paper [9], in this system it additionally uses sound sources height data, if this height of the sound source is greater than a specified threshold then it results unlikely to be a fall activity. In this approach, the falls detection rate due to background noise will be reduced. This system gives a good overall performance of 3% false detection rate with 100% of fall detection rate.

The acoustic sensor can be combined with a floor sensor to design a fall detection system, this combined system will be more robust for fall detection. In this paper [12], they built a fall detection system with the combination of sound sensing and floor vibration sensors. From sound and vibrational signals they have extracted the temporal and spectral features, using these extracted features they classified the fall and non-fall activities using Bayes' classifier, In this

system they performed experiments on a doll that mimics a human, to simulate fall activities, this system gave an overall 97.5% detection rate and small 1.4% of false detection rate.

It may appear like non-computer based techniques are suitable for wide range of applications in the field of fall detection systems, in practically they had several problems; the non-computer fall detectors are inconvenient (They have to be wear by the user all the time) and they will easily effected by the noise in surroundings (floor vibration sensors and acoustic sensors) . For these non-computer vision techniques additional hardware (data acquisition card) is also required to convert and obtain the analogue signals to digital signals so that they can be processed in PCs, these leads to additional costs. To overcome the non-vision based problems vision based fall detection techniques were introduced. In the vision based fall detection techniques, there is no necessity of wearing any equipment's/sensors and it is less effected by the surrounding/background noises: these cheaper in cost, only low cost USB cameras are enough for the computer based techniques due to this vision based fall detectors are economical.

2.3 Computer vision based methods

In the past 12 years, the computer vision and image processing techniques have been advanced to large extent. The tracking of subject moment in real time based on vision opened up a field of methods for fall detectors. In comparison with non-computer vision based techniques the computer vision based techniques have so many advantages such as: 1. these are convenient, user does not need to wear any kind of sensors or equipment's such as accelerometer sensor; 2. These are less effected by the noise in the surroundings (unlike acoustic sensors and floor vibrational sensors which are affected a lot). Based on the algorithms used for fall detections, these computer vision based fall detectors are categorized into two: analytical methods and machine learnings methods.

2.3.1 Analytical methods

In analytical methods, first some certain types of features will be extracted from the vision/video data, afterwards these features will be analyzed for the determination of fall and non-fall activities. The threshold based algorithms are the most used and popular analytical method, in threshold based algorithms, first the threshold will be set by empirically, afterwards to determine fall or non-fall activities these extracted features will be compared with the thresholds. It determines whether the action is a fall or non-fall.

In these papers [13] and [14], they have stated a fall detection system consists of a computer server and an Omni-dimensional camera, this has no blind spots because it captures an image of 360 degrees covering whole place simultaneously in a single shot. In this system, first a clean background will be obtained (image without subject). Afterwards, the current image will be subtracted from the background image as a result we will obtains foreground of interest. Noise will be removed from the image, afterwards a rectangle will be created that it encloses the object.

The height and width of these rectangle will be extracted from these aspect ratio (height to width ratio) will be obtained and compared with a certain threshold value to determine fall or non-fall activity. This threshold value is customized depending upon the personal physique. This system gives an overall detection rate of 90% with personal data and a detection rate of 78% without personal data.

In this paper [15], they have stated a fall detection system based on the changes in persons shape and motion history image. By using the frame differencing method the motion history image will be obtained (MHI), from this motion history image the movement amplitude will be extracted. Whenever this movement amplitude changes abnormally, the changes in persons shape information (such as changes in angle of enclosing ellipse and changes in aspect ratio of enclosing rectangle) will be compared with thresholds to determine fall and non-fall detection. The thresholds will be obtained empirically, this system gives a good performance rate of fall detection with acceptable rate of false detection with 87.5% of specificity and 88% of sensitivity will be obtained.

In this paper [16], they have stated a fall detection system based on the features of human shape deformation along the video sequence. Using shape context matching along the video sequence the silhouette of the person will be tracked, and the procrustes shape analysis will be applied on the extracted values of the contest matching results from this they have estimated the shape deformation values. These shape deformation value will be compared with the threshold for determination of fall and non-fall activities detection, by analyzing the mean and standard deviation values of shape deformation with in a time period an inactivity period will be detected. They applied ROC (curve) in this experiment under the optimal threshold values this system gives an overall of 96.4% of specificity and 95.5% of sensitivity.

In this paper [17], they have stated a threshold based novel context fall detection system in real time environment. In this system the image will be divided into small parts/blocks, firstly they obtains the centroid location between head and feet, the centroid location will be used to obtain a context model that consists head and floor (feet) parts. For every floor block, a Gaussian distribution of head blocks was trained it is used to find the mean location of the head (mean of heads blocks centroid positions). For the detection of fall head and feet will be located in image. Based on the vertical distance between head location and mean location of head in the floor block to feet location. If this distance is more than certain threshold value, then a fall will be detected. This system have achieved a good fall detection rate of 96% in a real world environment. The heads location is the core of this system, if the head is invisible the system will fails.

In some works, in the place of 2D features 3D features were extracted, and to detect fall and non-fall activities they were compared with certain threshold values. In paper [18], they have stated a similar approach in this they obtained 3D bodies of subjects. Initially, they obtained foreground region by using background subtraction method with a depth-image. By using a special camera TOF (time of flight) they obtained the 3-D position of the foreground regions centroid, the TOF camera was a self-calibrated by automatic floor detection. Distance between the 3-D locations of the foreground region centroid to the floor plane will be extracted. Later this distance will be

compared with certain threshold value to detect fall activity. This system gives an overall good performance, in experimental part under 0.4m of height threshold, it obtained the fall detection rate of 100% with a minimal 2.7% of false detection rate.

In this paper [19], it stated another similar approach of threshold based technique on extracted 3-D features. To reconstruct 3-D shape of the person it uses calibrated cameras. This is based volume distribution, it extract the volume distributed along the vertical axis to detect the fall. The fall will be detected when this volume distributions major part is distributed close to the floor for a defined time period, this implies that person has fallen. This system gave a good performance results (it achieves fall detection rate of 99.7% or more with four or more cameras), in this GPU (graphic processing unit) was used for efficient computation.

The threshold based algorithms overall performance is totally depends on the specified threshold values. Proper threshold values should be chosen carefully to obtain a good performance fall detection rate.

2.3.2 Machine learning algorithms

As stated in paper [20], Machine learning algorithms were been used in a wide range of areas and lot of developers have used these machine learning algorithms for the applications of fall detectors. In machine learning algorithm, from vision data different types of video features will be extracted, classifiers will be trained from these extracted video features. The classifiers will be trained either by supervised methods (like support vector machine, neural network, k-nearest neighbor and Hidden Markov Model (HMM)) for the classification of the different activities or postures for the fall detection or unsupervised methods (like mixture of Gaussians, single Gaussian, one class support vector machine and Parzen window) to determination of falls and normal activities.

a. Supervised methods

In supervised methods based fall detection systems, from short video sequences or postures the video features will be extracted. To detect falls or activities or different postures a supervised classifier will be trained from the extracted video features. In these papers [21] and [22], they have stated fall detection based on posture recognition algorithm. In these initially background subtraction method is used to obtain the segmented human body region. Afterwards from these segmented human body region the projection histogram features will be extracted. From these extracted features of the projected histograms, they will train different supervised classifiers such as poster probabilistic template stated in paper [22] and neural fuzzy stated in paper [21] for the classification of different postures. In case the posture is changed from stand posture to lie posture in small interval of time [21] or for a long interval of time the posture remains in lie posture [22], it reports a fall activity. In paper [21] a 97.8% of posture recognition was obtained And it showed four results of poster recognition, to show that posture recognition combined with a rule set detects a fall successfully (if posture changed from stand to lie in short time fall will be

confirmed), and a optimized posture recognition gives an fall detection rate of 90% for different data sets reported in paper [22], in this fall will be detected if posture will be in lie state for long time.

b. Unsupervised methods

As mentioned in paper [20], an unsupervised method will can find the unlabeled data hidden structure or normal model that has unlabeled data follow. Feature extracted from short video clips and postures these data can be collected from specified subject or human from their normal daily activities real time video stream and the extracted data can be used to build a activity model by using the unsupervised learning methods, which furthered developed for the detection of falls and in general activities. In these papers [22] and [23], they have used the unsupervised learning fall detection method. In this paper [23], to obtain video data they used only one camera and to track the human in this video they used the particle filter method with an enclosed ellipse technique. They extracted position data and normal activities of the persons from the tracking results. These extraction features are used in obtaining the “normal activity region” by EM (expectation maximization) technique. When person’s position is outside of the “normal activity region” for longer than a specified time threshold value, it detects that activity as a fall.

In this paper [24], persons tracking will be carried out by shape matching method. Person’s silhouette will be tracked through of the video sequence. From this silhouettes shape context data will be extracted and based on this extracted shape context data shape deformation will be calculated. A Gaussian model will be build based on the calculated shape deformation information, this model can detect the normal activities of the person. Afterwards this model can determine fall and non-fall activities. By using multiple cameras this model can obtain a good performance. Based on the experimental results, by using four cameras a 100% of perfect fall detection and false detection 0% can be obtained. In unsupervised learning methods a specified person solution will be obtained so there is no necessity of data collection from a lot of people to train a fall detection classifier. But a of lot time will be taken for constructing this specified person fall detection model to collect enough data from that specified person and until this data collection finishes, falls will not be detected.

2.4 Literature Survey Matrix

Comparison of non-computer vision based methods and computer vision based methods is given in below table (see Table 2.1).

Methods Description	Equipment	Merits	Demerits
Non-computer vision based methods	User-worn accelerometer devices, sound or vibration sensor	No infringement of personal privacy	High rate of false alarm; inconvenient to wear, easily affected by background noise
Computer vision based methods	Single or multiple digital cameras	Cheap, convenient to use and high accuracy	Some infringement of personal privacy, and the performance is strongly linked with illumination level

Table 2.1: Comparison of non-computer sensor and computer vision based methods

The proposed system compared with all the existing computer vision based methods work from literature survey as shown in table (see Table 2.2).

Journals	Equipment	Angle of coverage	Image processing techniques	Fall detection Parameters
Our system	Single camera	180	Background subtraction, Silhouette, contours, tracking and Human detection	Aspect ratio, orientation angle, travelled distance, vertical velocity and height
13 and 14	Omni dimensional camera computer server	360 degrees	Background subtraction model	Rectangle (aspect ratio)
15	Single camera	180	Motion history(MHI) Frame differencing	Rectangle (Aspect ratio) and ellipse (orientation angle)

16	Single camera	180	Silhouette (both foreground and edge detection)	Procrustes is a tool to quantify the human shape
17	Single camera	180	By tracking head, head, mean location and feet	Distance (Head and feet)
18	TOF camera	3D	Background subtraction on depth image	Distance Centroid of the person and floor plane
19	Calibrated (Multiple) cameras with GPU usage	3D	Silhouette approach	Volume distribution along the vertical axis, with certain time period
21 and 22	Single camera	180	Hog feature extracted from the human body region from background subtraction	Based on posturers, lies for certain time interval fall
23	Single camera	180	Particle filtering for tracking	Outside of activity region for certain time
24	Multiple Cameras	360	Silhouette	Shape deformation with time periods

Table 2.2: Literature Survey

2.5 Summary

In this chapter, some previous fall detection methods based on both non-computer vision techniques and the computer vision technique were reviewed. Our system is based on vision technique, it uses camera for information. For the non-computer vision techniques, some non-computer vision sensors (acceleration sensor, floor vibration sensor and acoustic sensor) were used to capture the sound, vibration and human body movement information for fall detection. The non-computer vision based methods are used widely in the fall detection field; however, they are either inconvenient to use or easily affected by noise in the environment, unlike our system. Our system is easy and convenient to use, also our system doesn't effected by noise that much compare to non-computer visions based methods, because in our method we will use effective filters to remove the excess noises.

Our system is based on the computer vision methods, normally computer based techniques uses one or multiple cameras are applied to extract the video information, in our system we will use single camera to extract the information. Our system is an optimized thresholds model for fall detection. Unlike other classifier based models our systems doesn't need large data sets and classifiers, due to this our system takes less computational time and no need of costly hardware.

The current fall detectors are posture classification based methods, these are not easy to distinguish fall activities from lying activities. But our system can easily distinguish fall and sitting, by combining the posture, tracking and velocity parameters. In lot of fall detection systems they won't detect humans, so they counts the non-human (object) falls as human falls, this will leads to false readings, but our system detects the humans and avoids false fall detection. For most of the threshold-based methods, only one particular parameter (such as the ratio between the fitted rectangle or 3D head velocity) is used for comparison with the threshold, which is not enough and may generate high false detection rate. In our robust fall detection system, different types of features (combination of ellipse angle, aspect ratio, velocity, vertical velocity of highest point and minimum height) will be extracted, and a decision will be made by comparing above features with proper thresholds in an organized way. Our system overcomes these problems.

CHAPTER 3 METHODOLOGY

This section summarizes the methodology used in this study for developing of a real time camera based human fall detection. The methodology involves five subsections: Workflow, hardware specifications, connections, working and Diagnosing the disease

3.1 System Overview

The human fall detection system that we proposed, in this thesis, is a low-cost computer vision based device, it's consists of an embedded computer (Raspberry Pi 3 Model B) and camera (Raspberry Pi camera module, 5 Megapixel, 1080p resolution). (See Figure 3.1) shows our system, that raspberry pi connected with camera module. This system can be setup on walls or ceilings, it will starts detection without any human interference.



Figure 3.1: Raspberry pi 3 interface with camera module

3.2 Flowchart

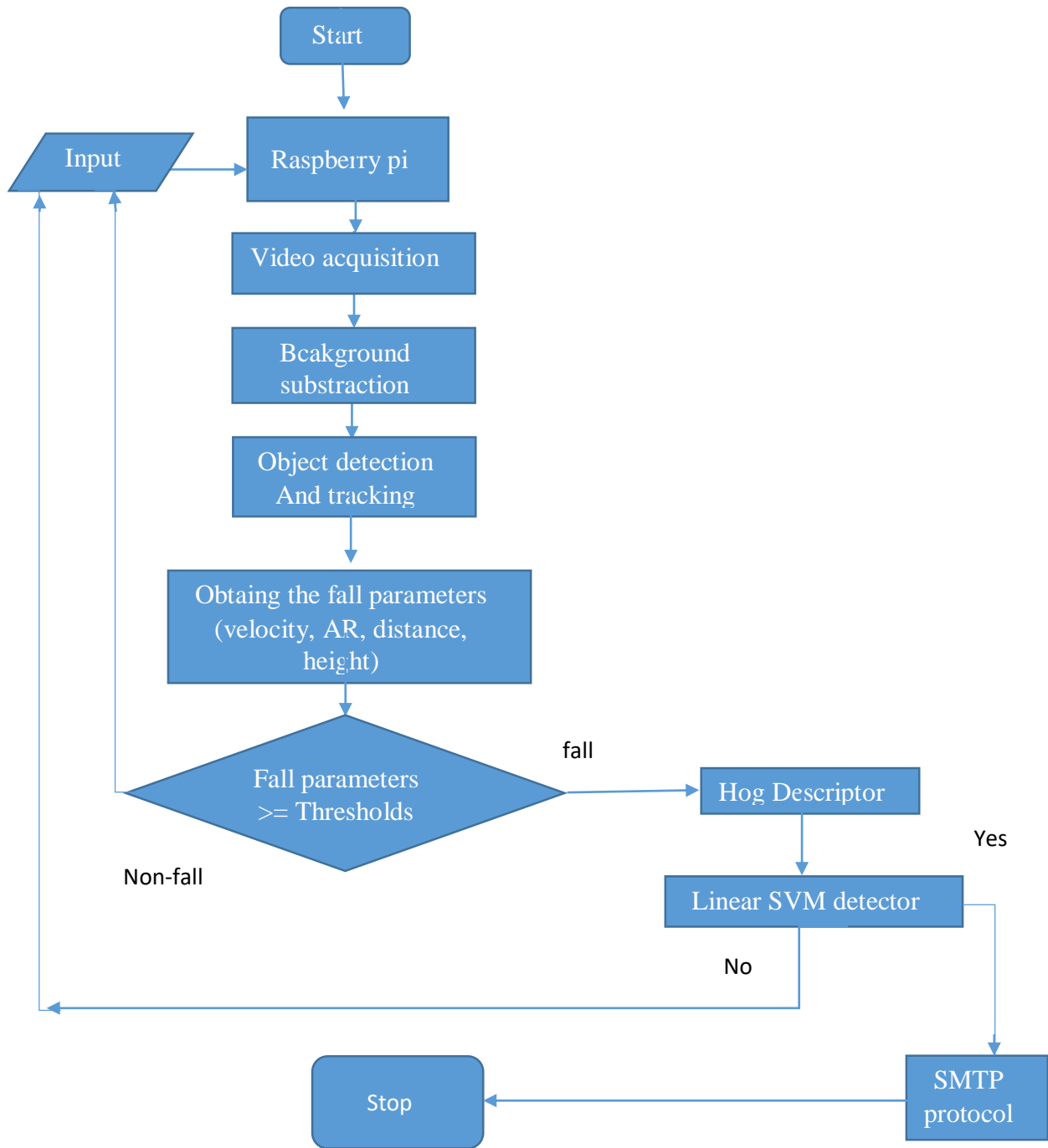


Figure 3.2: Flow chart of system working

The flow chart explains the working of the system in real-time as explained in figure (see Figure. 3.2). From camera module real time input video will be given to the raspberry pi, then raspberry pi performs different image processing techniques on the input image and extracts the background, silhouette and detects the object (elderly people). Based on algorithms we will obtain the fall parameters such as vertical velocity, aspect ratio, traveled distance and change in height. Afterwards rule based classification will be done from this fall and non- fall events will be obtained. In non-fall case system will stops the processing and again starts processing of new frame. In fall case the image will be processed through HOG (Histogram of Oriented Gradients) and linear SVM (Support Vector Machine) model are used for the detection of human, if human is detected it will further processed to SMTP protocol and sends email to respected take care personals. If human is not detected then it will stops the processing and starts processing new frame.

3.3 Hardware specifications

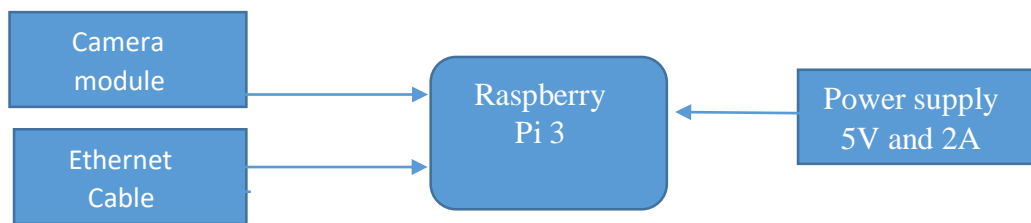


Figure 3.3: Hardware block diagram

As shown in figure (see Figure 3.3), the hardware setup consists of Raspberry Pi 3 Model B and camera (Raspberry Pi camera module, 5 Megapixel, 1080p resolution). 5V and 2A of power supply is given to Raspberry pi through adaptor. There will be only two connections is hardware setup an Ethernet cable is connected to Ethernet port of the raspberry pi and camera will be interfaced on raspberry pi using cable. The cable will be connected to the collector which is in between HDMI port and Ethernet as shown in below figure (see Figure 3.4).



Figure 3.4: Interfacing camera module to raspberry Pi

For internet interface we don't need any enabling, internet will be enabled automatically. But for camera we need to do it manually. After interfacing the camera module in raspberry pi for the first time, we have to enable the camera option, for this we go to run command "sudo raspi-config" it will display various control options as shown in below figure (see Figure 3.5), afterwards navigate to the interface and enable the camera option in it.

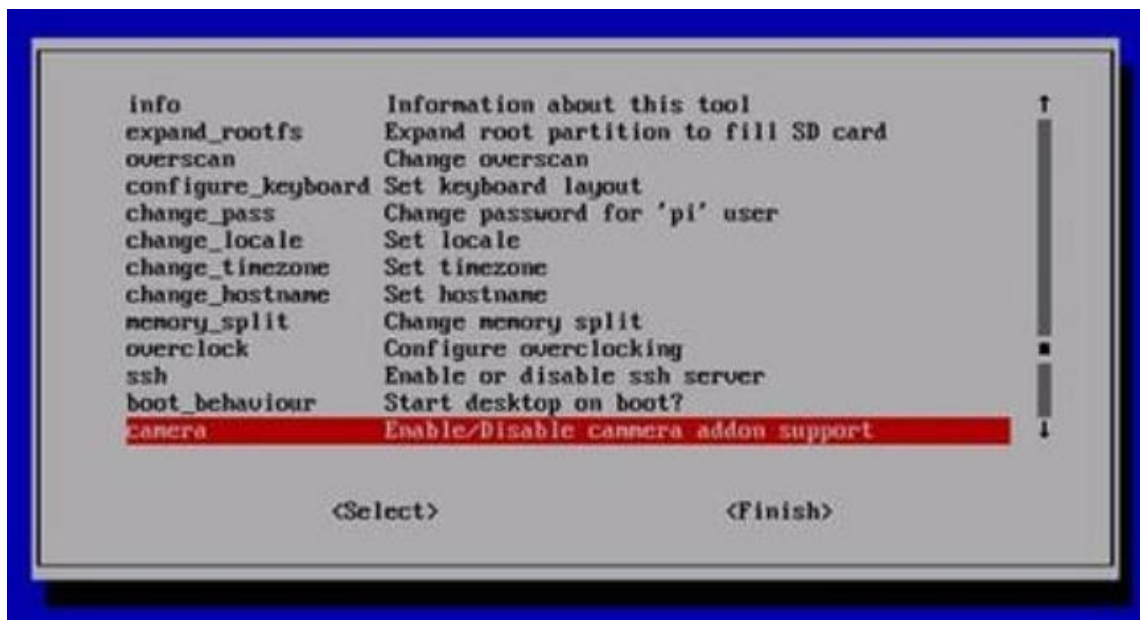


Figure 3.5: Enabling camera interface in raspberry Pi

3.4 Software setup

- Installation of Raspbian operating System on Raspberry Pi.
- Installation of Virtual Network Computing (VNC) server on the raspberry pi. It shares the duplicate raspberry pi display screen to the VNC viewer.
- Installing VNC viewer on my Personal Computer (PC). In my PC using VNC viewer I can view and control the raspberry pi. (Instead of using Monitor, keyboard and mouse).
- Obtain the Raspberry Pi (Internet Protocol) IP address to connect to VNC sever. By default username will be pi and password will be raspberry.
- Installation of OpenCV (Open Source Computer Vision) library. OpenCV library mainly for real-time computer vision techniques. Installation of OpenCV in Raspberry Pi is a very lengthy process and it takes a lot of time (Up to 12 Hours and more).

3.5 Image Processing Techniques

3.5.1 Video acquisition

We will obtain the real-time input video with a resolution of 640 x 480 pixels and at a frame rate of 30 FPS (Frames per Second) from the camera module. But technically image operations on a video of 640 x 480 pixel at 30 FPS in Raspberry Pi is too computational. We cannot change the FPS, because lowering or increasing of FPS will lowers the performance of the system. So to optimize the system we will resizes resolution of the video to 320 x 240 pixels.

3.5.2 Background subtraction

The video frames obtained from the camera will be in BGR (blue, green and red) color-space. We will convert the BGR images into GRAY images (intensity images), this conversion reduces the processing time by 2/3. The gray scale image will be smoothed with median filter [25] to reduce the ambient noise present in the images. Median filter is good for removal of small and moderate noises, and also it preserves the edges.

Background subtraction requires less computational power so it is well suited for real-time (live streams). In general, we can simple state background subtraction as method of comparing pixels values in current image with a reference image to obtain the differences between those two images. Pixels in the current frame are compared with background reference frame, if they are similar they treated as background pixel, if different then it's a foreground pixel.

In our system, for reference frame we runs the video sequence that doesn't contains any moving objects for some time (until 50 frames) and we will consider the last frame in this sequence as

reference frame. Because sometimes camera starting errors can occurred so that we will end up with a fault reference frame.

Background subtraction algorithm is as follows

$$\mathbf{I}(\mathbf{x}, \mathbf{y}) = |\mathbf{I}\mathbf{C}(\mathbf{x}, \mathbf{y}) - \mathbf{I}\mathbf{R}(\mathbf{x}, \mathbf{y})| \quad (3.1)$$

In equation (3.1), x and y are horizontal and vertical coordinates. Where $I_c(x, y)$ and $I_r(x, y)$ are intensity of pixel in current frame and reference frame. The background subtracted image of our system is shown in figure (see Figure 3.6).



Figure 3.6: Background subtraction

3.5.3 Object detection

Objects that are present in current frame can be detected by thresholding the above background subtracted image with a threshold value T based on equations (3.2) and (3.3).

$$\mathbf{I}(\mathbf{x}, \mathbf{y}) > T \quad \longrightarrow \quad \mathbf{I}(\mathbf{x}, \mathbf{y}) = 1 \quad (3.2)$$

$$I(x, y) \leq T \quad \longrightarrow \quad I(x, y) = 0 \quad (3.3)$$

After applying the equations (3.2) and (3.3) on background subtracted image we obtain a black and white (binary) image. In our experiment the T is 15. In this binary image objects are represented by white pixels as shown in figure (see Figure 3.7).

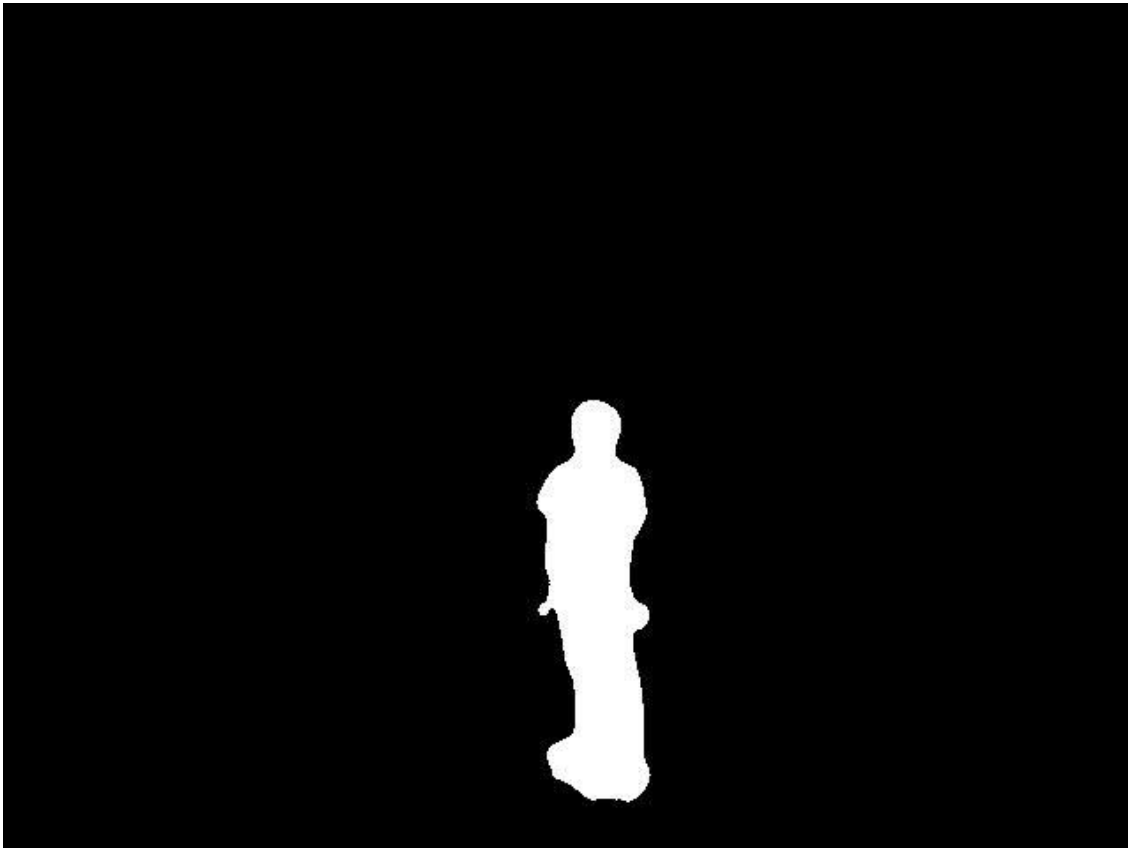


Figure 3.7: Binary Image from the differenced image

3.5.4 Morphological operations

As we can see in above figure (see Figure 3.7) there are some holes (black dots) were appearing inside the detected objects. To fill or remove these kind of gaps/holes we will use the morphological operations. Morphological operations can only be performed on binary images. They requires two inputs, they are input binary image and kernel or structuring element. Structuring element describes the nature of the operation. The morphological operations that were used in our system are dilation, erosion and closing operations.

a. Erosion Technique

Erosion technique is similar to soil erosion process, it erodes (thins) the boundaries of the detected object. The erosion kernel slides through the binary image (in 2D). If all the pixels in the kernel are 1 then only it keeps that pixel value as 1, if any value of pixel under kernel is zero then it makes that value as zero.

So basically is that, whenever we applies the erosion technique the boundaries of the objects will be discarded (removed) based upon the kernel size. This erosion technique removes small white noises in images. Erosion Technique showed in figure (see Figure 3.8).

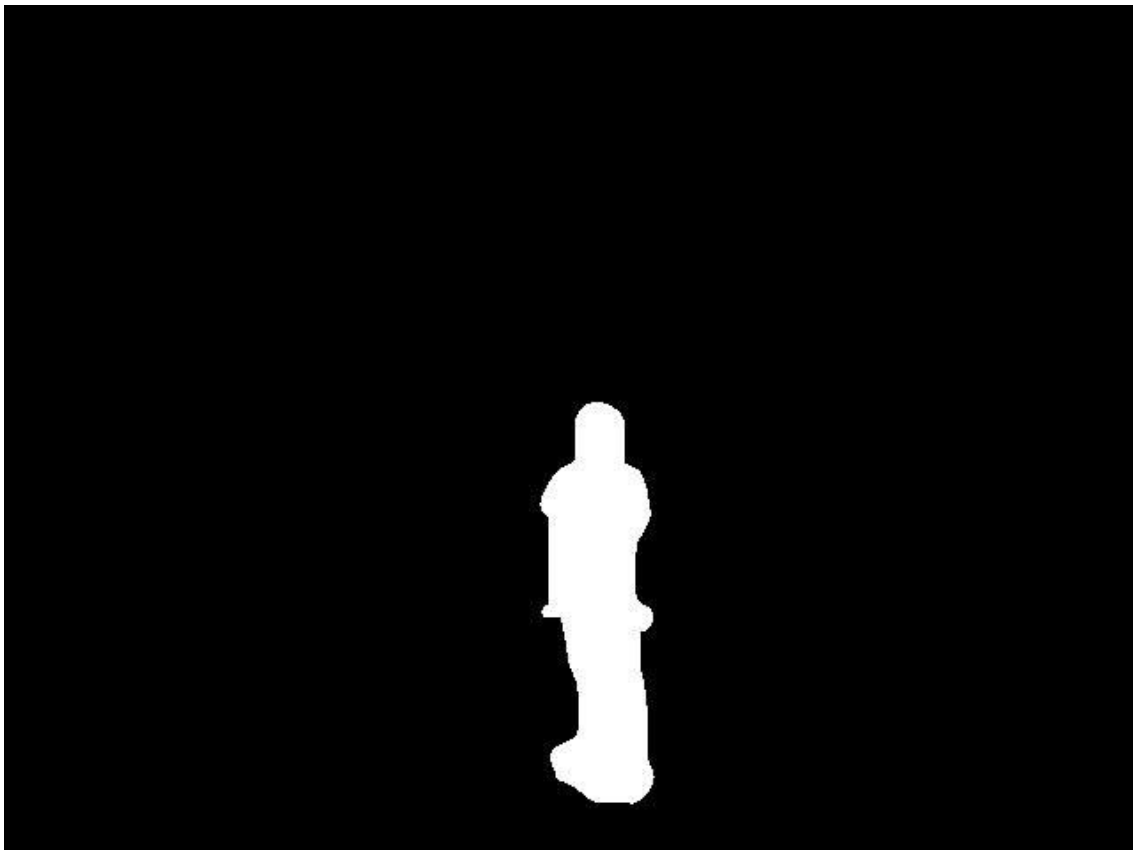


Figure 3.8. Binary image after Erosion operation

b. Dilation Technique

Dilation Technique is the exact opposite of erosion technique. In this technique even on of the pixel value under kernel is 1 then it keeps that pixel value as 1. So in this case the size of the object will be increased at the boundaries. Basically, in conditions like noise removal, after erosion we will do dilation. Because, once erosion is done for noise removal the object will shrinks based on erosion kernel size, so if we applies dilation with same kernel size as erosion

we can increase our object area to normal without noise. Erosion Technique showed in below figure (see Figure 3.9).

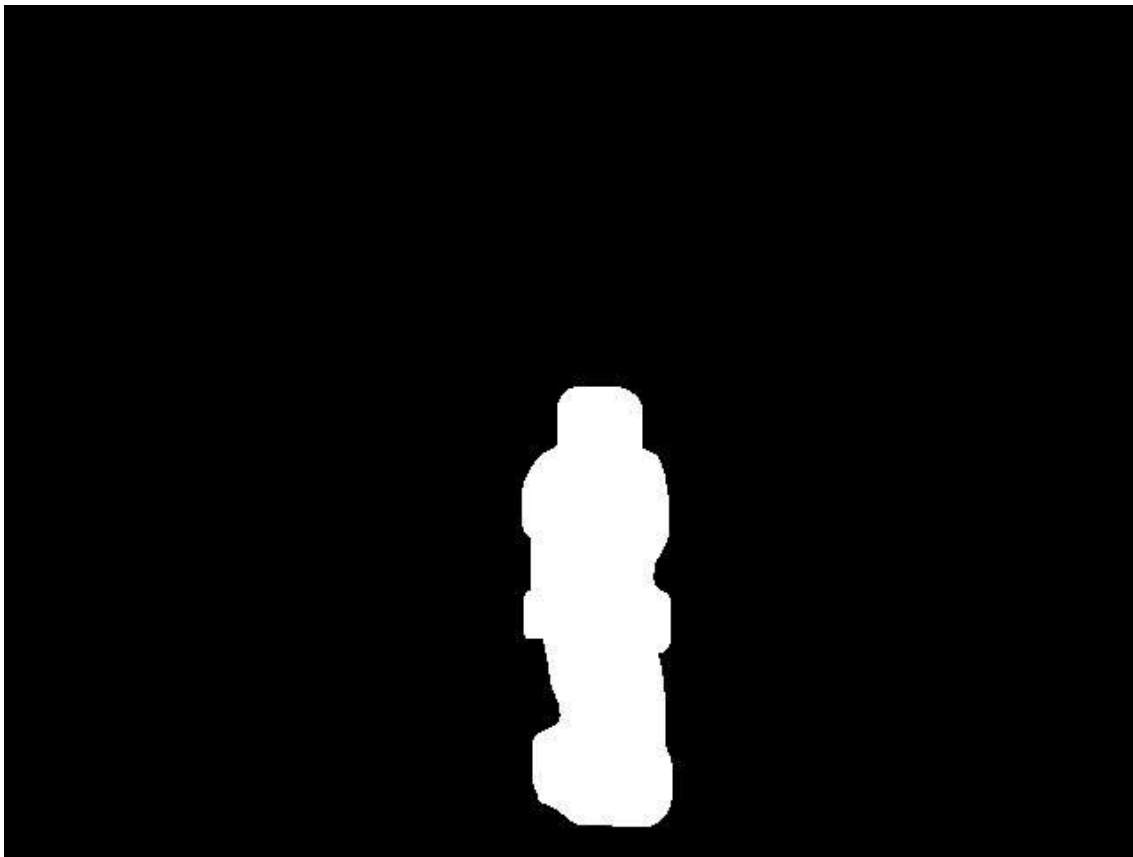


Figure 3.9: Binary image after Dilation operation

c. Closing Technique

Closing technique is the combination of dilation technique followed by erosion technique. Closing technique is useful in closing small holes in the object.

In our system we will use only closing technique instead of using erosion and dilation separately. Closing technique showed in figure (see Figure 3.10).



Figure 3.10: Binary image after Closing operation

3.5.5 Tracking of the object

Contours

Contours can be defined as curve that joins all the continuous points (throughout boundaries) that have the same features (color or intercity). We will apply the contour technique on the resulted binary image after morphological operations to obtain contours. Basically finding contours is same as finding white objects from our binary image. These contours are used for object recognition, detection and analysis.

Tracking

We will draw a bounding rectangle using contours of the detected object. Bounding rectangles encloses the object in it and touches the boundaries of the object from four sides. The bounding rectangles are two types as follows:

- a. Straight bounding rectangle
- b. Rotated rectangle.

Straight bounding rectangle is a straight rectangle, it doesn't depend on the rotations of the object. Rotated rectangle tries to cover up object with in minimum area, so it depends on the orientation of the object. So it consists of angle of rotation additionally along with its other coordinates. So straight bounding rectangle is error free for small (minor) camera rotational errors.

We will track the highest point of the object from the ground. For this we will draw the straight bounding rectangle on the object. We can obtain all four corners of the straight rectangle. The coordinate of the corners are (x, y) , $(x + w, y)$, $(x, y + h)$ and $(x + w, y + h)$ respectively. In digital image processing top left corner acts as origin $(0, 0)$, so all the coordinators starts from top left corner of the frame. As shown in below figure (see Figure 3.11).

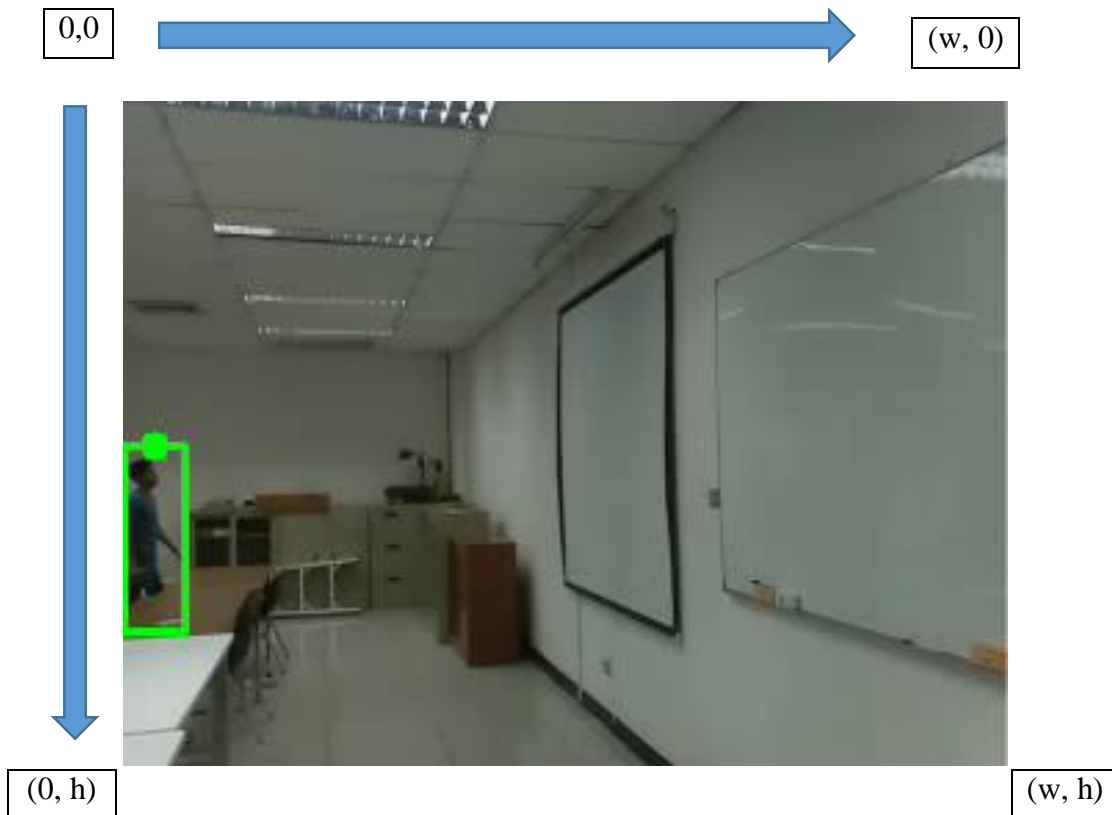


Figure 3.11: Coordination system of digital image processing

As shown in the figure (see Figure 3.11), It is clear that the coordinates of highest point of the object will be $(x + w/2, y)$, the bounding rectangle we drawn is a straight bounding box rectangle so the highest point of the object is independent of rotations made by object. We will track this point motion by frame to frame approach to obtain the fall detection parameters. The tracked point can see in below figure (see Figure 3.12) green dot on rectangle.



Figure 3.12: Tracking point (green dot)

3.6 Fall detection parameters

3.6.1 Velocity of the object and traveled distance

The fall detection algorithm that used in our system mainly depends on velocity of the object. It will be calculated based on the changes of our tracking point (highest point of the object). We will find the velocity by frame to frame approach (consecutive frames). To calculate velocity we need time and distance parameters.

The time difference is the time taken between current and previous frame as shown in the equation (3.4). It will be in between 0.0833 to 0.1 seconds. Distance can be obtained simple by subtracting current coordinators with previous coordinators as shown in the equation (3.5). With respect to previous coordinators these changes are a straight deviation in both axes.

$$\mathbf{Time} = T_c - T_p \quad (3.4)$$

$$\mathbf{Distance} = \sqrt{(x_c - x_p)^2 + (Y_c - Y_p)^2} \quad (3.5)$$

In expression (4), T_c is the current time and T_p is the previous time. In expression (5), (X_c, Y_p) are the current point coordinators and (X_p, Y_p) are the previous point coordinators in X and Y axes. In figure (see Figure 3.11) the point X and Y coordinate are (14,160).

The distance can be calculated as the distance travelled by the tracking point from the two frames divided by the time difference of those two frames. As shown in expression (6), D_c and D_p are current and previous distance values. When fall occurs the velocity will moves faster before it reaches the ground it will be 12.5 pixels/second. Normally it ranges from 0-10 frames per normal activities.

$$\text{Velocity} = \frac{D_c - D_p}{t_c - t_p} \text{ Meter/second} \quad (4.5)$$

3.6.2 Aspect Ratio (AR) and angle

Using the contours we will draw a straight bounding rectangle and a fitting ellipse that encompasses the detected object. The aspect ratio is the division of height and width of the straight bounding rectangle [26]. Angle of the object is the angle between ellipse long axis and image horizontal plane [26]. With these we can easily differentiate the sitting and falling activities. As we stated aspect ratio is the object's height over its width, if aspect ratio is lesser than one and the angle of object is in between 40-140 fall will be detected. If AR greater than 2.6 and angle 165 to 195 or 0 to 15 then it will be stands state, if AR is 1.4 to 2.1 and angle is 195 to 220 and 15 to 40 it will be sitting state. In below figures (see Figure 3.13, 3.14 and 3.15) standing, sitting and falling states were detected based on AR and angle



Figure. 3.13: Standing state detection with AR and angle



Figure 3.14.: Sitting state detection with AR and angle



Figure 3.15: Fall state detection with AR and angle

3.6.3 Vertical Velocity

In our algorithm vertical velocity is the main fall detection parameter. As the name suggested it only calculates the changes of the tracking point in vertical (along y-axis) direction. In this we will calculate vertical velocity with a constant time interval C_t . Here the distance travelled by tracking point will be the vertical distance (Change of height). The change of height will be calculated for every consecutive frame it's the difference between current heights to previous height. The vertical height will be calculated as change in height to the time taken in between C_t . C_t is the difference between current time (t_c) and to initial time (t_i). The vertical velocity equation is:

$$V = \frac{1}{t_c - t_i} \int_{t_i}^{t_c} h'(t) dt \quad (3.6)$$

In equation (3.6), $h'(t)$ is the height derivative of tracking point with respect to time. In our system Ct will be 0.65 and V is 8.5 As we stated earlier $t_c - t_i = Ct$ is a constant time interval.

3.6.4 Height

In previous section we saw how much height is important in calculation of vertical height apart from that individually height can also be used as a fall detection parameter. The tracking point is the highest point of the object, its height value can be used to avoid false alarms. The final height of the fallen person cannot be more than one meter. The threshold height will be 100 pixel that is almost half to person's height.

3.6.5 Thresholds

So after extracting or calculating all of the above parameters, individually we will keep thresholds to every parameters (AR, angle etc) to detect the falls. And we will make a fall condition which is a combination of all the fall parameters. Slightest changes of these parameters threshold results changes the total outcome (Accuracy, precision and recall) of the system, so these thresholds are obtained carefully from experimenting falls in real time.

3.7 Human detection

A real time human detection is a challenging task, it requires lot of processing. In our algorithm we will use region of interest (ROI) which will be obtained by foreground estimation. We will obtain the Histograms of Oriented Gradients (Hog) features from the obtained ROI. State Vector Machine (SVM) will detects the non-detected and detected human regions by classifying the Hog features. Further human detected regions will be filtered by using the area of overlapping regions. This proposed human detection algorithm is eight time faster compare to original multi scale HOG.

3.7.1 ROI selection

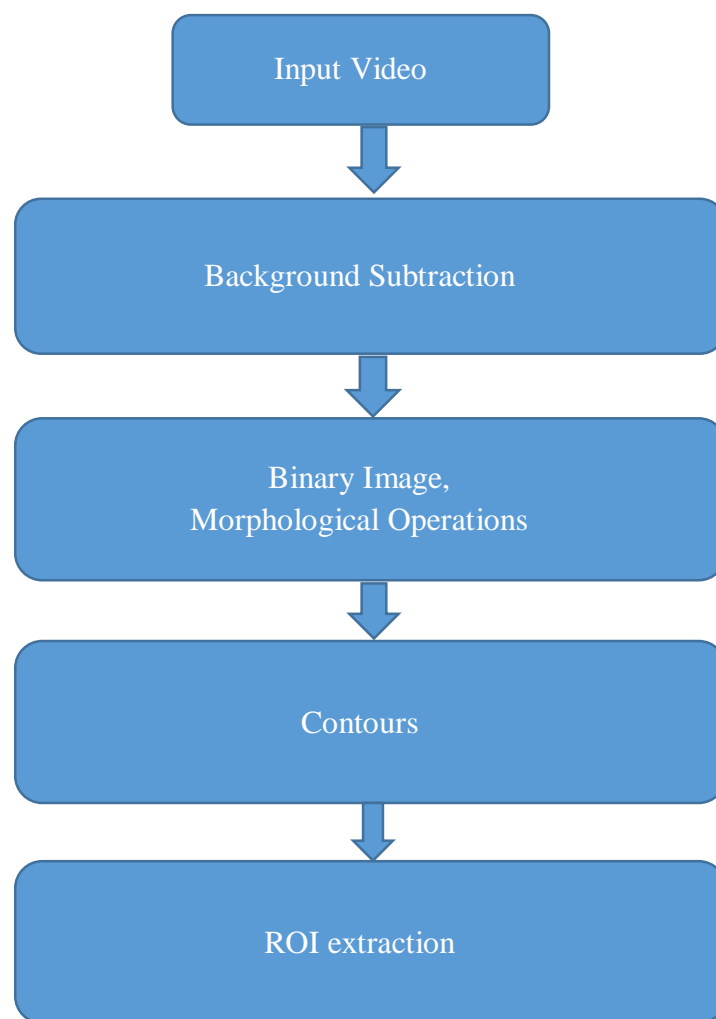


Figure 3.16: ROI extraction from input video

Processing a video in real time for human detection on Raspberry Pi is too computational. If we use ROI technique the processor will works only on certain amount of region on the frame instead

of processing whole frame. To obtain ROI we have to estimate the foreground, we have already obtained foreground from the background subtraction method in section 3.5.2. AS explained before, to find coordinates of the only foreground we will make it to binary image, morphological operations, finding counters and we will draw straight bounding rectangle on it. The bounding rectangle that enclosed whole object it itself called region of interest (ROI). ROI extraction can be shown in figure (see Figure 3.16).

3.7.2 Human detection using Hog

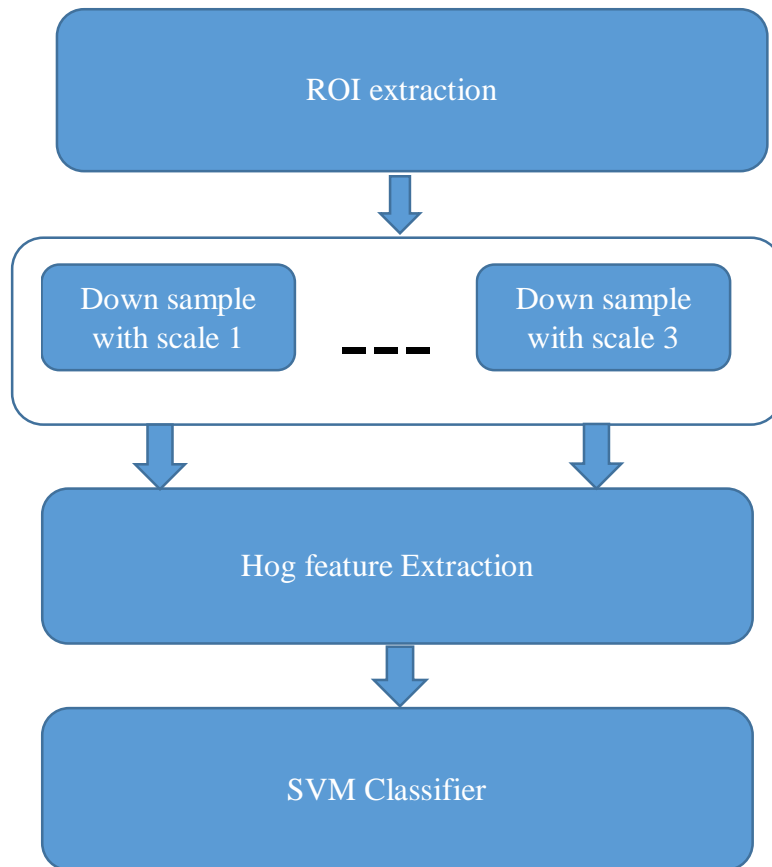


Figure 3.17: Human detection using HOG and SVM

Due to extraction of Region of Interest (ROI) the computational cost of the human detection algorithm will be reduced. As shown in above figure (see Figure 3.17) after the ROI extraction, we will apply the (HOG) Histogram of Gradients features on ROI for multi-scale human detection.

For HOG feature extraction we will obtain the gradients of the image first, using sober filter in both x and y directions. The magnitude and the phase of the gradient will be calculated with equations (3.7) and (3.8):

$$\text{Magnitude (S)} = \sqrt{s_x + s_y} \quad (3.7)$$

$$\text{Phase } (\theta) = \arctan(s_y/s_x) \quad (3.8)$$

After obtaining the magnitude and phase, now they will be arranged in nine bins (boxes) based on their orientation (0 - 180) degree with magnitudes acts as weights for that orientation box, this process is called Orientation binning it is show in below figure (see Figure 3.18). Hog will contains 8x8 cell size so to obtain a histogram, all of the pixels contribution in the cells (8x8) will be summed up to obtain the 9-bin histogram. This process can be shown as:

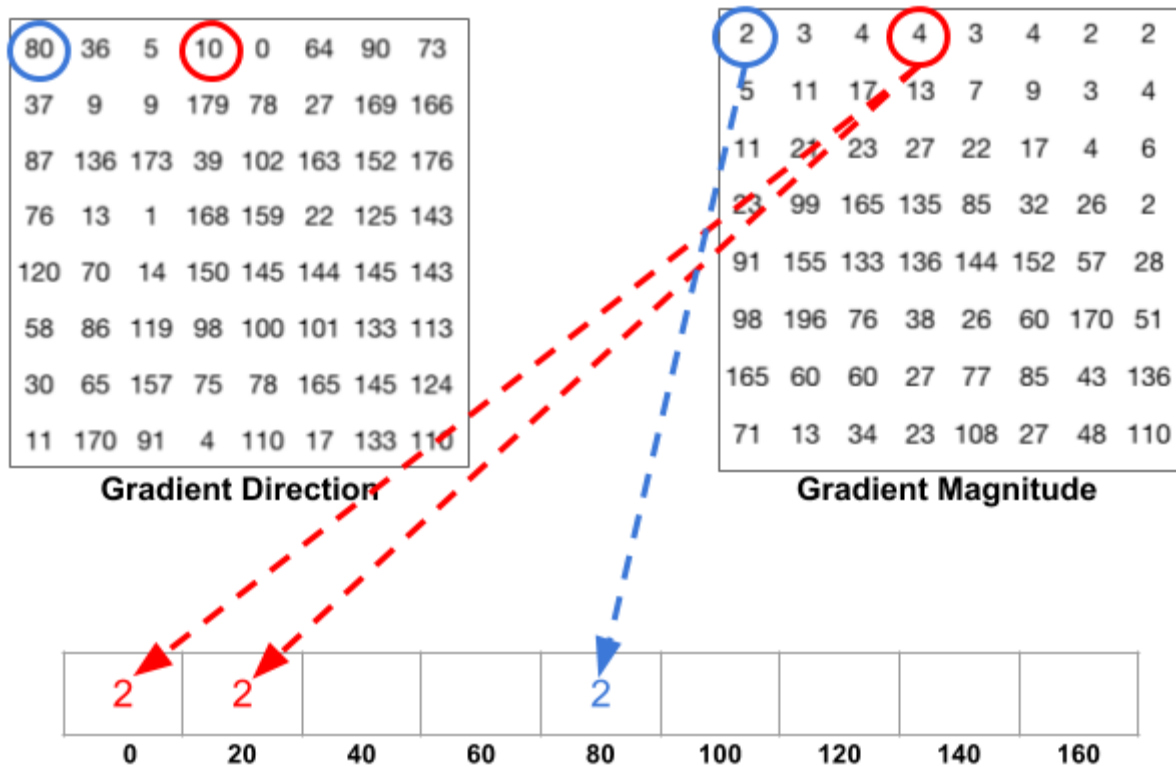


Figure: 3.18: Orientation binning

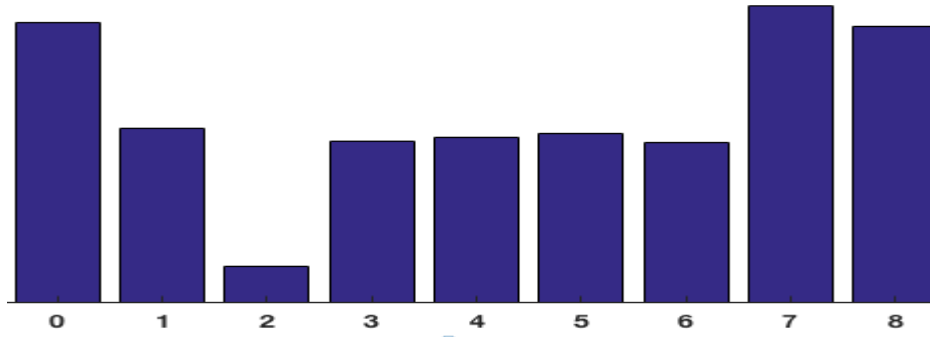


Figure 3.19: Histogram of Gradients (8x8 cell)

As stated, all of the pixels contribution in the cells (8x8) will be summed up to obtain the 9-bin histogram shown in figure above figure (see Figure 3.19).

For the extraction of HOG feature will be carried out with a cell size of 8x8, block size of 16x16 and nine orientation bins (placed evenly). As shown in below figure (see Figure 3.20):

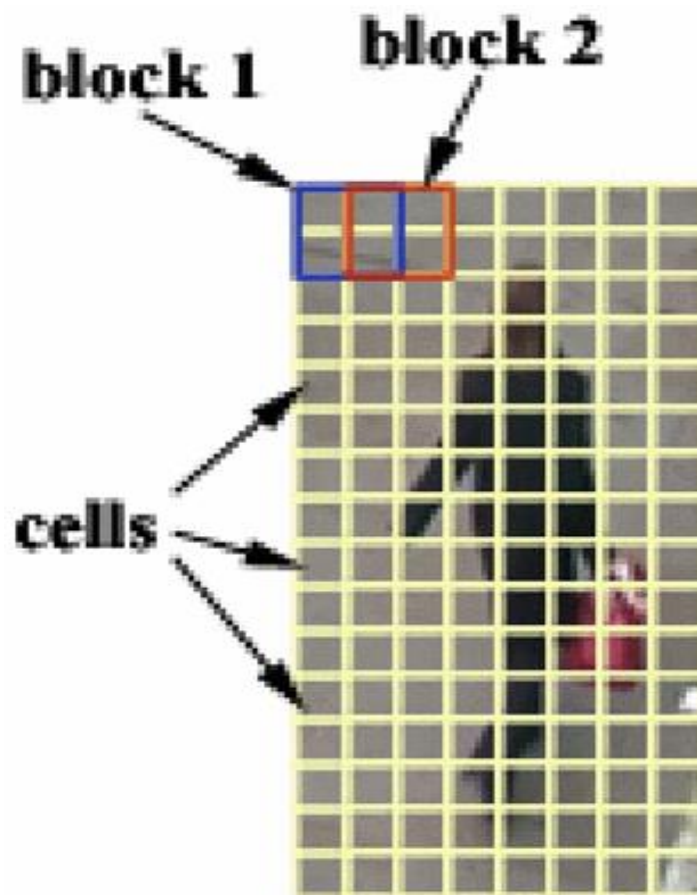


Figure 3.20: Cells and blocks structure of HOG

After extracting the HOG features, now we will classify the computed features using a linear support vector machine (SVM). The computed HOG features will be given to linear SVM that is trained with INRIA person detection dataset. The SVM uses hyper plane to separate the two classes, the result out of this case is may be the detector window contains the person or it doesn't contains the person.

The detector window will slides all over the image to detect humans. In some cases objects size may be bigger than detector window size (when person is so near to camera). To detect persons bigger than detector window, the image will be down sampled with a certain factor. This factor is called scale stride (ss).

Single scale detection: If detection is done on image without changing its resolution (no need of down sampling) it's called single scale detection.

Multi-scale detection: If an image decreases its resolution based scale stride (ss) for detection of human in the image is called as multi-scale detection. In multi-scale detection, first feature extraction and classification will be done on whole image with its original size $W \times H$ (width and height). Then image will be down sampled due to this the image size will becomes $DW \times DH$ (Down sampled width and height respectively). $DW = W/SS$, $DH = H/SS$ and Updates Scale stride $SS = ss \times 1.05$. Down sampling can be seen in figure (see Figure 3.21). The detection will be performed on resized image, after this the image will again under goes to down sampling and detection will processed, this will continues until the image size becomes smaller than that of detector window size. In the end all the detection results of scales will be combined. For raspberry pi multi-scale with scale stride of 1.05 takes lot of computational time, in fact, if we uses multi scale we can't obtain real time. To avoid these we will use a maximum of three scale detection and it will have trade-off the accuracy and computational time

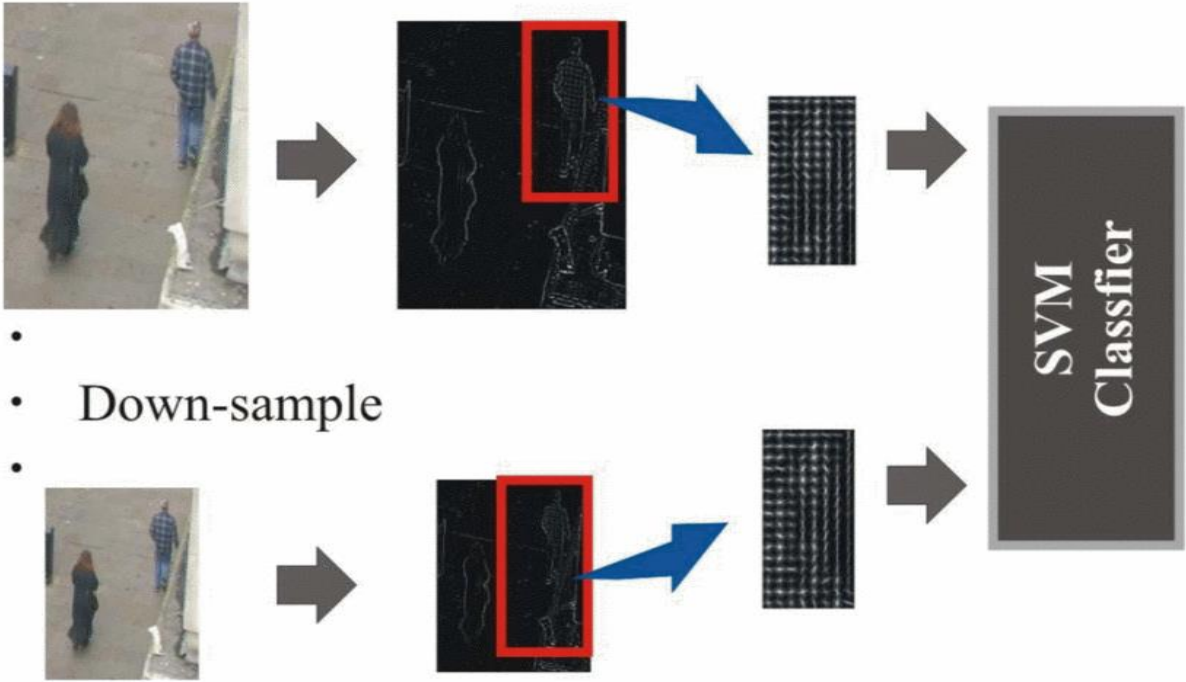


Figure 3.21: Down sampling of the image

The detector window sizes will be 48 x 96 (for upper region of the image) and 64 x 128 (for whole image) as showed in figure (see Figure 3.22). We are using 64 x 128 size window for up to two scale levels and 48 x 96 size window will be used with one scale.

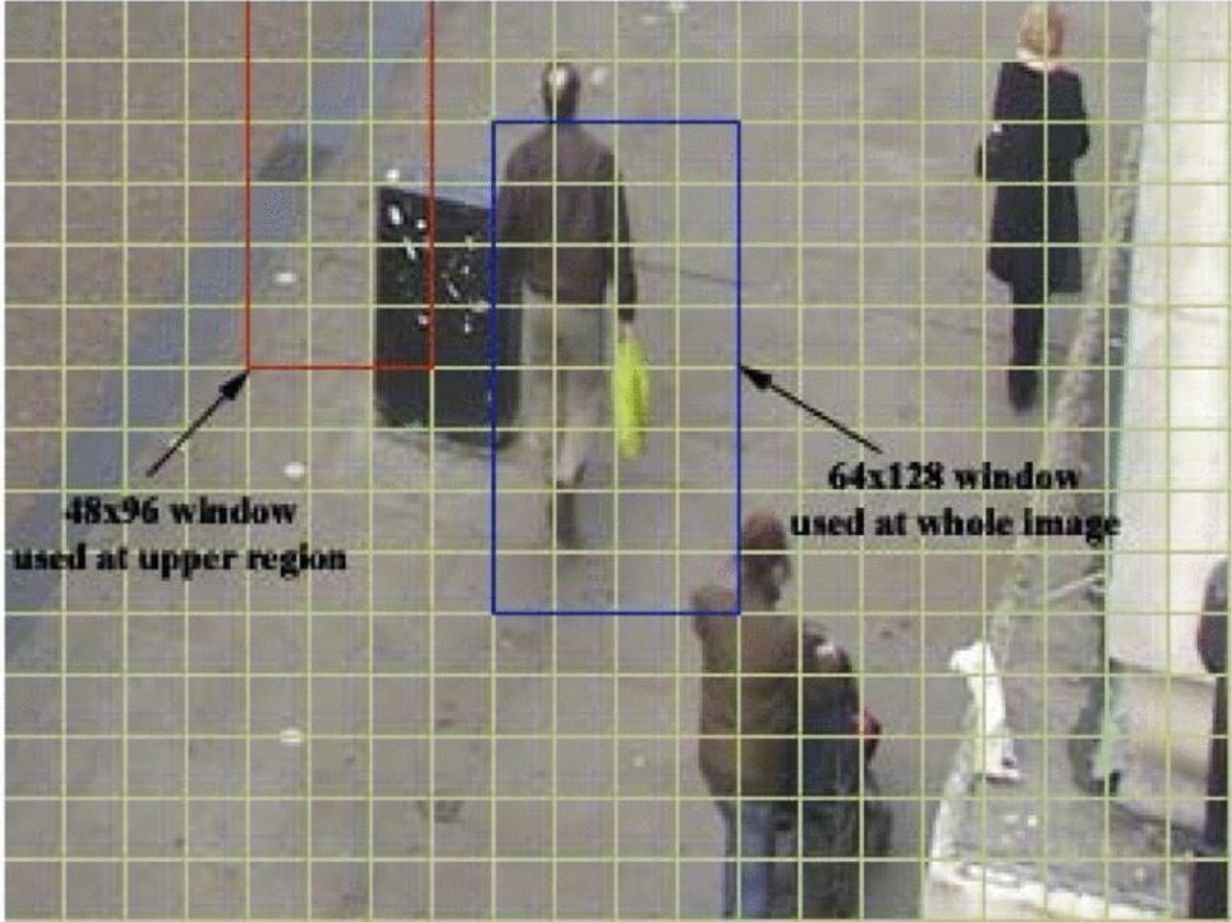


Figure 3.22: Detector Windows

3.7.3 Filtration of Detection

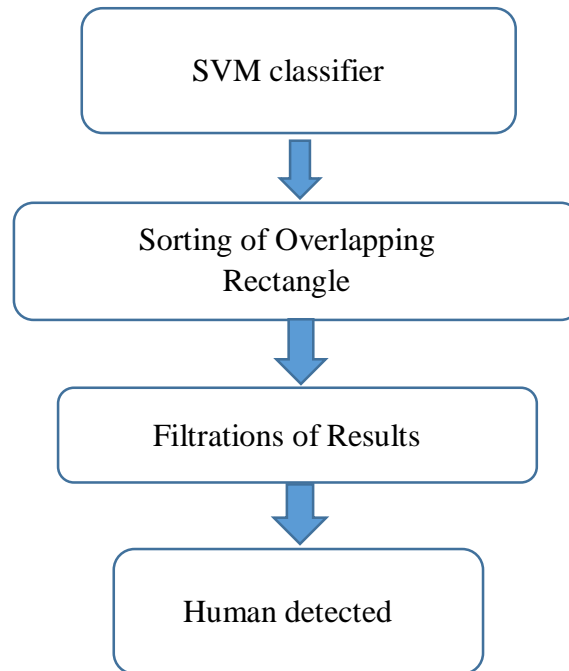


Figure 3.23: Filtrations of detections results

Human may be detected at the one scale image or they may detected in second level and third level like-wise. In the result image we will obtain more than one rectangle's, so the results should be filtered so that we will obtain only one rectangle on person. For this we followed a simple approach by keeping mind about raspberry pi processing power. First we will find the areas of each individual rectangle, that is width times height ($w * h$), then we will find the overlapping rectangles area. If the overlapping of rectangles is more than a threshold value then one of them will removed by equation:

$$(A1 \cap A2) > T, (A2 \cap A3) > T, (A(N-1) \cap AN) > T \quad (3.9)$$

In equation (9), T is the threshold value, ($A1, A2 \dots\dots An$) are the areas of rectangles, N is the number of rectangles. For single person the threshold will be greater than sixty percentage. The final result of the human detection will be shown in figure (see Figure 3.24).

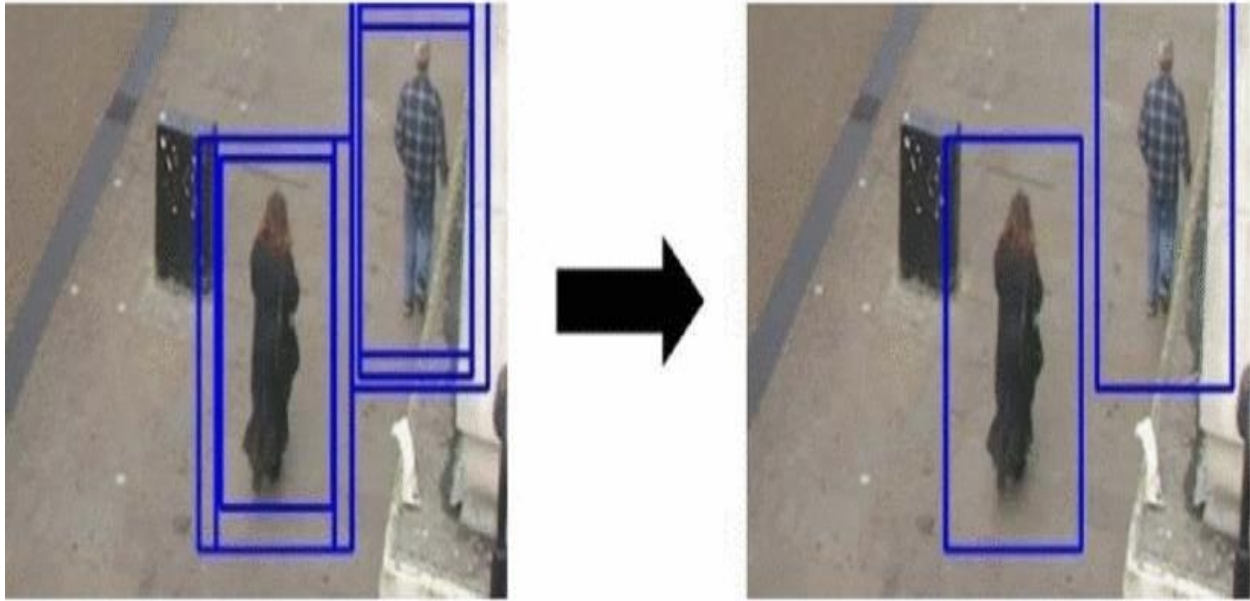


Figure 3.24: Elimination of overlapping rectangles

3.8 Simple Mail Transfer Protocol (SMTP)

SMTP is an internet's standard host-to-host mail transfer protocol. To send email through SMTP we will install two packages (ssmtp and mailutils).

The ssmtp package, it is a light-weighted SMTP server used to send email, SMTP itself stands for simple Mail Transfer Protocol. We will also install mailutils (it's a set of libraries for handling emails). After the installation we will first edit the ssmtp configuration file (ssmtp.conf file) using sudo command, in configuration file we will set mail hub to Gmail at port 587, we will give our Gmail user ID and password and we will enable the encryption.

As shown in the below figure (see Figure 3.25), we will import the SMTP library (smtplib) and we will specify our email and password and the recipient address. The email body will consist of a subject, header, and body. Body is our original message, in our case, it's fall detected. Subject is the subject or main header of the email and finally header consists of information of to whom we are sending email from which email ID and subject at the end. We will access the SMTP with port 587, we will be logged into our mail and an email will be sent with the sendmail command, in the end we will close the server with the quit command.

```
import smtplib

smtpUser = 
smtpPass = 

toAdd = 'updown@down@hotmail.com'
fromAdd = smtpUser

subject = 'Python Test'
header = 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'Subject: ' + subject
body = 'From within a Python script'

print header + '\n' + body

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

s.ehlo()
s.starttls()
s.ehlo()

s.login(smtpUser, smtpPass)
s.sendmail(fromAdd, toAdd, header + '\n\n' + body)

s.quit()
```

Figure 3.25: Source code to send email using SMTP protocol

CHAPTER 4 EXPERIMENTAL RESULTS

In our system, data will be recorded through camera in real time, camera will be placed at any of the top corners of the room so, it can cover the whole body of the human. For the detection of systems performance, a total of 48 real time videos were recorded in a closed class room. In this, 14 videos are only the straight falling videos, 8 videos consists both sitting and falling activities and remaining 26 videos consist of depict falls in different directions. Human fall detections are performed on this all video in real time.

So when a fall is occurred the fall detection parameters changes abnormally that is they will reach threshold falling conditions, hence the fall will be detected by our system as shown in below figure (see Figure 4.1).



Figure 4.1: Raspberry pi monitor display of fall detection

An email will be sent when the fall is detected as shown in below figure (see Figure 4.2). We have created a new email id: falldetectionpy@gmail.com to send mail (ssmtp.conf contains password so it's not secure to use personal mails), in this email the subject is the assistance request for patient 143 and main body is state of the person fallen.

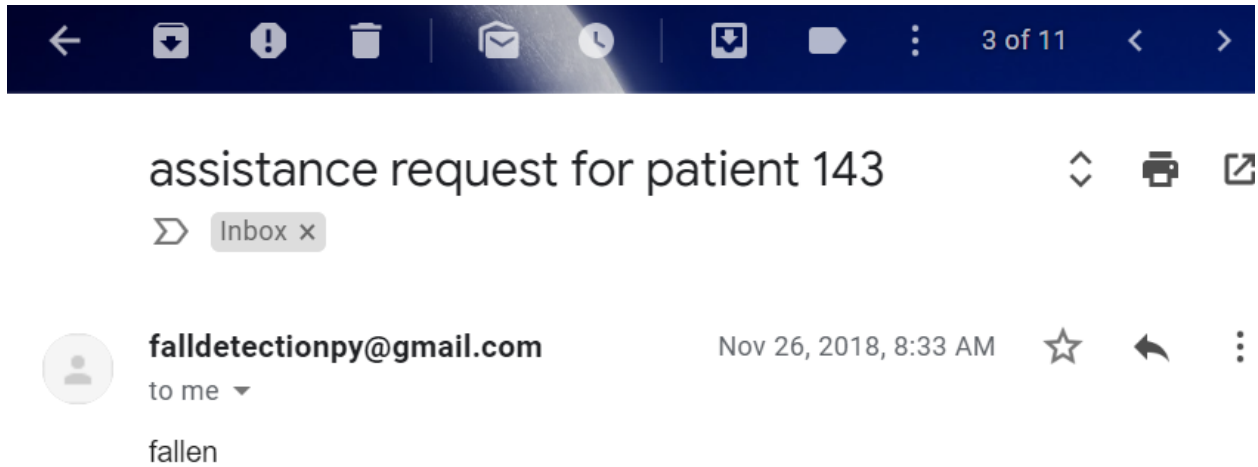


Figure 4.2: The Alert email sent to recipient when fall detected

4.1 Calculation of Accuracy

The accuracy of this detection is calculated by forming a confusion matrix. Confusion matrix has four main parameters and according to this case they can be defined as:

TP (True Positive): In this case it is the predicted yes (fall detected) and it is a fall.

TN (True Negative): In this case it is predicted no and there is no fall.

FP (False Positive): Predicted yes, but actually there is no fall.

FN (False Negative): Predicted no, but actually there was a fall .

Table 4.1: Experimental results obtained in real time

		Predicted condition	
		True Positive 15	False Positive 1
Actual condition	Fall		
	Non-fall	False Negative 2	True Negative 14

For the above confusion matrix (see Table 4.1) accuracy, recall and precision is calculated as:

To calculate Precision from Table 4.1.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (4.1)$$

$$\text{Precision} = \frac{15}{(15 + 1)} \times 100$$

$$\text{Precision} = 93.75\%$$

TP (True Positive) and FP (False Positive).

To calculate Recall from Table 4.1.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (4.2)$$

$$\text{Recall} = \frac{15}{(15 + 2)} \times 100$$

$$\text{Recall} = 88.23\%$$

TP (True Positive) and FN (False Negative).

To calculate Accuracy from Table 4.2.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \quad (4.3)$$

$$\text{Accuracy} = \frac{(15 + 14)}{(15 + 14 + 1 + 2)} \times 100$$

$$\text{Accuracy} = 90.65\%$$

TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative).

The overall precision, recall and total classification of accuracy is 93.75%, 88.23% and 90.65%.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This thesis goal is to develop a real time camera based human fall detection using raspberry pi. We successfully extracted the foreground features with background subtraction, and we track the highest point of the object by finding contours from threshold binary image. Using that tracking find we found the important parameters of the fall detection (vertical velocity, aspect ratio, angle, velocity and height). We built robust algorithm to obtain the fall detection and we even detected humans with less computational method by HOG in conjunction with SVM and finally whenever the fall occurred we sent an email with the SMTP protocol.

Compare to previous detection systems, our system is robust and reliable in real time, the false alarms rate is very low, unlike other systems we concerned both postures and velocities of subject, to build the reliable fall detection algorithm. We even verified whether the fall took was a human or non-human with ROI scheme. Over all our system is low cost, high performance, real time, reliable system with accuracy of 94.44% and sensitivity of 100% and specificity of 96.48%.

5.2 Recommendations

The main recommendations required are,

- The biggest recommendation is that increasing computational power of embedded system (raspberry pi).
- Moving cameras and using multiple cameras, for better tracking. Night vision fall detectors.
- Need new digital image processing techniques to find fall detection not only inside environment, for outside too.
- Personal data bases, these eliminates false alarms.

REFERENCES

- [1] World Health Organization. WHO Global Report on Falls Prevention in Older Age. World Health Organization; Geneva, Switzerland: 2007.
- [2] Moiz Ahmed, Nadeem Mehmood, Adnan Nadeem, Amir Mehmood and Kashif Rizwan."Fall Detection System for the Elderly Based on the Classification of Shimmer Sensor Prototype Data". Korean Society of Medical Informatics, Health inform Res. 2017.
- [3] MINGOzzI, ENzO, and Claudio Cicconetti. "Enabling Technologies and Standards for Mobile Multihop Wireless Networking." *Mobile Ad Hoc Networking: Cutting Edge Directions*, Second Edition (2013): 34-76.
- [4] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, 2006.
- [5] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait and Posture*, vol. 28, no. 2, pp. 285–291, 2008.
- [6] M. Estudillo-Valderrama, L. Roa, J. Reina-Tosina, and D. NaranjoHernandez, "A proposal of a fall detection algorithm for a multidevice personal intelligent platform," 8th IEEE International Conference on Bioinformatics and Bioengineering, Athens, Greece, 2008.
- [7] M. Estudillo-Valderrama, L. Roa, J. Reina-Tosina, and D. NaranjoHernandez, "Design and implementation of a distributed fall detection system-personal server," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 874–881, 2009.
- [8] Y. Li, Z. Zeng, M. Popescu, and K. Ho, "Acoustic fall detection using a circular microphone array," 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Argentina, 2010.
- [9] B. Mungamuru and P. Aarabi, "Enhanced sound localization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 34, no. 3, pp. 1526–1540, 2004.
- [10] Y. Li, K. Ho, and M. Popescu, "A microphone array system for automatic fall detection," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 1291–1301, 2012.

- [11] M. Popescu and A. Mahnot, "Acoustic fall detection using one-class classifiers," 31st Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA, 2009.
- [12] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.
- [13] S. Miaou, F. Shih, and C. Huang, "A smart vision-based human fall detection system for telehealth applications," *The Third International Conference on Telehealth (IASTED)*, Anaheim, CA, USA, 2007.
- [14] S. Miaou, P. Sung, and C. Huang, "A customized human fall detection system using omni-camera images and personal information," 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, Arlington, VA, USA, 2006.
- [15] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), Niagara Falls, Ont., Canada, 2007.
- [16] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Procrustes shape analysis for fall detection," *ECCV 8th International Workshop on Visual Surveillance (VS 2008)*, Marseille, France, 2008.
- [17] M. Shoaib, R. Dragon, and J. Ostermann, "View-invariant fall detection for elderly in real home environment," *Fourth Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, Singapore, 2010.
- [18] A. Leone, G. Diraco, and P. Siciliano, "An automated active vision system for fall detection and posture analysis in ambient assisted living applications," *2010 IEEE International Symposium on Industrial Electronics (ISIE)*, Bari, Italy, 2010.
- [19] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-d silhouette vertical distribution," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 290–300, 2011.
- [20] C. Bishop, "Pattern recognition and machine learning," Springer, 2006.
- [21] C. Juang and C. Chang, "Human body posture classification by a neural fuzzy network and home care system application," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 37, no. 6, pp. 984–994, 2007.

- [22] R. Cucchiara, A. Prati, and R. Vezzani, "An intelligent surveillance system for dangerous situation detection in home environments," *Intelligenza Artificiale*, vol. 1, no. 1, pp. 11–15, 2004.
- [23] H. Nait-Charif and S. McKenna, "Activity summarization and fall detection in a supportive home environment," *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, Cambridge, UK, 2004.
- [24] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 611–622, 2011.
- [25] Gonzalez, "Digital image processing," 3 Ed., Prentice Hall, 2008.
- [26] Vinay Vishwakarma, Chittaranjan Mandal, and Shamik Sural. Automatic detection of human fall in video. In *International conference on pattern recognition and machine intelligence*, pages 616–623. Springer, 2007.