Master Thesis
Electrical Engineering with
emphasis on Signal Processing
September 2016

Weapon Detection In
Surveillance Camera Images

Submitted By
Rohith Vajhala
Rohith Maddineni
Preethi Raj Yeruva

Department of Applied Signal Processing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
This thesis is submitted to the Department of Applied Signal Processing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Electrical Engineering with Emphasis on Signal Processing.

Contact Information:
Author(s):
Rohith Vajhala
E-mail: rova15@student.bth.se
Rohith Maddineni
E-mail: roma15@student.bth.se
Preethi Raj Yeruva
E-mail: prye15@student.bth.se

Supervisor:
Irina Gertsovich

University Examiner:
Dr. Sven Johansson

Department of Applied Signal Processing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Now a days, Closed Circuit Television (CCTV) cameras are installed everywhere in public places to monitor illegal activities like armed robberies. Mostly CCTV footages are used as post evidence after the occurrence of crime. In many cases a person might be monitoring the scene from CCTV but the attention can easily drift on prolonged observation. Efficiency of CCTV surveillance can be improved by incorporation of image processing and object detection algorithms into monitoring process.

The object detection algorithms, previously implemented in CCTV video analysis detect pedestrians, animals and vehicles. These algorithms can be extended further to detect a person holding weapons like firearms or sharp objects like knives in public or restricted places.

In this work the detection of weapon from CCTV frame is acquired by using Histogram of Oriented Gradients (HOG) as feature vector and artificial neural networks performing back-propagation algorithm for classification.

As a weapon in the hands of a human is considered to be greater threat as compared to a weapon alone, in this work the detection of human in an image prior to a weapon detection has been found advantageous. Weapon detection has been performed using three methods. In the first method, the weapon in the image is detected directly without human detection. Second and third methods use HOG and “background subtraction” methods for detection of human prior to detection of a weapon. A knife and a gun are considered as weapons of interest in this work. The performance of the proposed detection methods was analysed on test image dataset containing knives, guns and images without weapon. The accuracy rate 84.6% has been achieved by a single-class classifier for knife detection. A gun and a knife have been detected by the three-class classifier with an accuracy rate 83.0%.

**Keywords:** Back propagation, histogram oriented gradients, gun detection, knife detection, sliding window, neural network.
Acknowledgements

We would like to express our appreciation to all the people who supported us on completing our thesis. We would like to express our gratitude to our thesis supervisor Irina Gertsovich for all the efforts that she has put in order to help us progress forward in our thesis. We would like to thank her for all the valuable suggestions that she has given us during our journey. We also would like to be obliged to Blekinge Institute of Technology for providing us with the resources and for helping us go forward during our thesis.
## Contents

Abstract i

1 Introduction 1
  1.1 Aims and Objectives of Work .................................. 3
  1.2 Research Questions ........................................... 3
  1.3 Overview of Thesis Work ....................................... 3
    1.3.1 Segmentation ............................................. 4
    1.3.2 Feature Extraction ....................................... 4
    1.3.3 Classification ........................................... 5
  1.4 Thesis Organization .......................................... 6

2 Related Work 7
  2.1 Weapon Detection in Images and Videos ...................... 8

3 Object Detection 11
  3.1 Segmentation .................................................. 11
    3.1.1 Sliding Window ........................................... 11
  3.2 Feature Extraction ........................................... 12
    3.2.1 HOG Feature Extraction ................................ 12
  3.3 Classification with Neural Networks .......................... 15
    3.3.1 Back-Propagation algorithm .............................. 16
    3.3.2 Ten-Fold Stratified Cross-Validation .................. 18
    3.3.3 Performance Evaluation of a Classifier ............... 18
  3.4 Neural Network for Classification of Single Class ........... 20
  3.5 Neural Network for Classification of Multiple Classes ...... 20
  3.6 Image Databases for Evaluation of Performance of Different Methods 20
    3.6.1 Image Database of Knife ................................ 20
    3.6.2 Image Database of Gun ................................... 23
    3.6.3 Image Database of Human ................................ 24

4 Implementation 26
  4.1 Knife Detection in the Original Image ....................... 26
    4.1.1 Segmentation by Using Sliding Window .................. 26
    4.1.2 HOG Based Feature Extraction for Knife ............... 26
4.1.3 Neural Network for Classification of Single Class (Knife)  
4.1.4 Detection of Knife in a Real Time Image  
4.2 Human Detection Before Knife Detection  
4.2.1 Human Detection Using HOG Method  
4.2.2 Human Detection Using Background Subtraction  
4.2.3 Knife Detection Surrounding Detected Human Region  
4.3 Knife Detection in Recorded Video  
4.4 Detection of Knife and Pistol  
4.4.1 Neural Network as Multi-Class Classifier  
4.4.2 Training a Multi-class Classifier  
4.4.3 Parameters for Neural Networks  

5 Results and Discussion  
5.1 Detection of Knife in Image  
5.2 Background Subtraction Method  
5.2.1 Background Subtraction  
5.2.2 Knife Detection After Background Subtraction  
5.3 Detection of Human Using HOG and Neural Network  
5.3.1 Segmentation  
5.3.2 HOG Representation of Human  
5.3.3 Neural Network for Human Detection  
5.3.4 Knife Detection After Detection of Human  
5.4 Performance Evaluation of Multi-Class Classifier  
5.5 Performance Evaluation of Neural Network Used for Knife Detection Using Cross Validation  
5.6 Calculation of Computational Time  
5.7 Discussion  

6 Conclusions and Future Work  
6.1 Conclusions  
6.2 Scope for Future Work
# List of Figures

1.1 Flow-chart of proposed solution. .................................................................................. 4
3.1 Implementation of HOG. ............................................................................................... 13
3.2 Rectangular and circular block descriptors. ................................................................. 14
3.3 Neural network. ............................................................................................................ 15
3.4 Positive and negative examples ................................................................................... 22
3.5 Training and testing samples for multi-class classifier. ................................................. 23
3.6 Positive examples for human detection. ....................................................................... 24
3.7 Negative examples for human detection. ..................................................................... 25
4.1 Receiver Operator Characteristics graph. ..................................................................... 29
4.2 Real life example images .............................................................................................. 30
4.3 Horizontal and vertical edges of human. ...................................................................... 32
4.4 Flow chart of background subtraction. .......................................................................... 33
4.5 Example showing knife outside detected image segment. .......................................... 34
5.1 HOG feature descriptor for knife. ................................................................................ 37
5.2 Decrease of MSE with increase in number of iterations for neural network training for knife detection. .............................................................................................................. 38
5.3 Confusion matrix for single class classifier ................................................................. 38
5.4 Direct detection of knife ............................................................................................... 39
5.5 Background and image frame ...................................................................................... 40
5.6 Background subtraction and canny edge detection output. ......................................... 40
5.7 The segments classified as "knife containing segment" by the classifier B are shown by three rectangular boxes. ................................................................. 41
5.8 Segmentation using sliding window .............................................................................. 42
5.9 Extracted HOG features for human detection ............................................................ 43
5.10 Results for INRIA database. ....................................................................................... 43
5.11 Human classified as positive by neural network. ....................................................... 44
5.12 Sliding window for knife detection. ............................................................................ 44
5.13 Final output of neural network. .................................................................................. 45
5.14 Confusion matrix of multi-class classifier. ................................................................. 45
5.15 Comparison of outputs of three methods. ................................................................. 47
5.16 Different examples where HOG might fail ................................................................ 49
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Confusion matrix</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Dataset division for multi-class classifier</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>Positive and negative training sets for human detection</td>
<td>25</td>
</tr>
<tr>
<td>4.1</td>
<td>Positive and negative training sets for knife detection</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Parameters of neural networks</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>Results of stratified 10 fold cross-validation</td>
<td>46</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Closed Circuit Television (CCTV) is one of the possible operational requirements to be considered in a wider security and safety aspects [1]. CCTV cameras are being installed almost everywhere in public places for providing security [2]. Main uses of CCTV surveillance are for providing security, deterrence (providing certain measures like proximity alarms, improved site design, lighting, intruder detection systems), crime investigation and reduction in insurance costs [3]. Deterrence effects of CCTV cameras vary for different crime categories and even from time to time [1].

CCTV footage is considered as one of the most important evidence in law enforcement agencies and courts [3]. Value of CCTV footage as evidence depends greatly on visual quality of cameras used. Therefore, there is a steep increase in installation of good quality CCTV cameras in public squares [4]. There were 1.8 to 4.5 million CCTV cameras used for surveillance in UK alone [5]. By the end of 2010, there were more than 50,000 cameras installed in Sweden [4]. These CCTV cameras are installed in various public places like stores, public transport systems, taxicabs, schools and financial institutions [4]. It made the public feel safer and reduced fear of victimization [1]. In Poznan (Poland), with the help of installation of 450 cameras, government was successful in decreasing street fights by 40% and drug cases by 60% [2].

As a result of increase in number of CCTV cameras used for surveillance, number of screens to be monitored by a CCTV operator increased a lot. As mentioned earlier, one of the most important applications of CCTV cameras is deterrence. Monitoring multiple screens simultaneously is stressful task for CCTV operator. Number of screens each operator can monitor will greatly depend on various human factors [2]. As number of screens to be monitored by each human operator increase, concentration of human observation on each screen simultaneously decreases drastically with time [6]. For a single operator it is quite challenging to monitor multiple screens simultaneously for prolonged time. According to [7] research says that concentration of CCTV operator oscillates at a rate of 83%, 84% and 64% after one hour of continuous monitoring of 4, 9 and 16
Chapter 1. Introduction

Therefore, there is an increasing demand in search for automated surveillance algorithms. The main aim of automated surveillance is to alert CCTV operator when there is a dangerous situation. A dangerous situation refers to a person or a group of persons attacking, creating fear or disturbances in public places with dangerous weapons like knives or guns. The concept of automated surveillance is possible with the help of incorporation of object detection algorithms from the field of computer vision and image processing into cameras. Usage of object detection algorithms in software processing of the video material in surveillance cameras was started in recent years and generally used in intelligent transportation system for traffic monitoring [8].

Until now, number of algorithms proposed to detect weapons from surveillance cameras is very small. Additionally these proposed algorithms detect only one kind of weapon i.e either gun or knife. The aim of the thesis is to find an object detection algorithm that can identify a knife in surveillance camera image. Though the algorithm cannot stop the occurrence of crime automatically, it could aid to surveillance camera operator that monitoring multiple screens by alerting him when a possible knife is detected.

Until 1985, usage of CCTV was limited to private spaces [1]. Hence there were no proper standards regarding CCTV camera installation requirements. Later, when CCTV usage expanded to public places and security agencies, proper standards were set as minimum requirements for surveillance cameras [3]. According to [3] surveillance video camera resolution should have a minimum resolution of 640x480 pixels, record 5 to 8 frames per second if the subject is moving at normal speed and 12 frames per second if the subject is moving quickly [3]. Wide variety of surveillance cameras are used depending on requirements like fixed view, pan-tilt-zoom, infrared sensor, image intensifiers and thermal cameras [9].

Implementation of the thesis was carried out on low resolution images, since CCTV images are of low resolution. By this, we could devote our thesis mostly to real time scenarios. Another objective is to design a multi class classifier that can detect multiple types of weapons like gun and knife held by a person instead of single type of weapon. In this thesis we proposed a knife detection algorithm based on HOG feature extraction and neural networks classification. Proposed algorithm was tested on real time images i.e detection of knife in the hand of a person where image resolution is 640 x 480 pixels. We extended our proposed method to detect knife in a recorded video. Later we designed the classifier to detect both knife and a gun in the image.
Chapter 1. Introduction

1.1 Aims and Objectives of Work

The aim of the thesis is to propose a method to detect a particular kind of a weapon carried by a person in a low resolution image such as CCTV image of resolution 640 × 480 pixels. The objectives to be achieved are:

1. Detection of a human in an image.
2. Detection of segment containing knife in the detected image segment containing human.
3. Extraction of features for each segment.
4. Design of a neural network based classifier to classify a single type of weapon versus a non-weapon.
5. Computation time required to detect a weapon directly in an image as compared to detection of a weapon after human detection.
6. Design a multi class classifier to detect two types of weapon: knives and guns.

1.2 Research Questions

1. How to detect an object in a low resolution image?
2. Can we reduce the computational time required to detect a weapon by limiting the Region of Interest (ROI) as compared to detection of a weapon in original image?
3. What classification technique can be used for detection of object in a low resolution image?

1.3 Overview of Thesis Work

In our thesis, knife detection was done using three methods. The flow chart of proposed solution is shown in figure 1.1. In the first method, knife was detected directly in the acquired image. Process of detection of weapon carried by a person in an image can be divided into three main stages. The following subsections describe shortly each of three stages:
Chapter 1. Introduction

Figure 1.1: Flow-chart of proposed solution.

1. Segmentation
2. Feature extraction
3. Classification

1.3.1 Segmentation

The size of a knife in the hands of a person can be considered small relative to the size of person. Hence, calculating global features (features computed for whole CCTV image) may not be useful to detect the knife in an image as the global features cannot describe features of knife separately from the rest of image. Therefore, to calculate local features (features calculated after segmentation of small images from whole image), image captured by camera should be divided into small segments. The process of segmentation of complete image into small blocks is done with the help of sliding window mechanism [10]. Size of the image segment i.e sliding window is considered to be 100 × 100 pixels, as size of the knife might not exceed 100 × 100 pixels in an image of 640 × 480 pixels captured by CCTV camera.

1.3.2 Feature Extraction

Various feature extraction algorithms were proposed in the field of computer vision. Different feature extraction algorithms compute unique features of an object of interest in an image. Features like colour, shape and texture not always can be considered in weapon detection. For example reflective property of knife blade makes knife detection based on colour features not feasible. First we considered
Harris key point detection as our feature descriptor but as knife contains very few corners, it was found to be less effective. After literature study, it was assumed that knife can be detected by finding a feature descriptor that is capable of estimating the approximate shape of knife. HOG feature descriptor was chosen for our application as HOG features can describe the edges of knife blade.

1.3.3 Classification

Classification of an image segment as weapon or non-weapon can be done with the help of artificial neural networks trained through supervised or unsupervised learning algorithms. In this project, back-propagation algorithm which is a supervised learning algorithm is chosen in feed-forward neural network for its simplicity. A neural network with two hidden layers is used. Neural networks are trained with different number of neurons in hidden layers and different sizes of training sets. These designed neural networks are evaluated using ROC (Receiver Operator Characteristics) graph. Neural network with 50 neurons in first hidden layer and 30 neurons in second hidden layer was found to detect knives with greater accuracy. For detection of single type of weapon such as knife, one output neuron is required. For detection of two types of weapons such as knife and pistol, output layer requires three output neurons.

In the other two methods, shown in figure 1.1, a segment with human was detected in the image first and knife was detected from or around the detected segment. In second method human was detected using HOG (Histogram Oriented Gradient) feature extraction and neural network. In the third method human was detected using background subtraction method. Former method was computationally slower than later but works for the images acquired for example, by CCTV camera mounted indoor or outdoor environment where the background was not constant. Later method was computationally faster compared to former but works well assuming that CCTV camera was mounted indoor and the background remains constant.

In the background subtraction method human is detected by subtracting the image from the reference background image and knife is detected surrounding detected human region. In second method segmentation of human was done using sliding window of $400 \times 200$ pixels. As CCTV cameras can be mounted on the ceiling we assume that human fits in $400 \times 200$ pixels segment. After detection of human in an image knife is detected in and surrounding the human segment.
Chapter 1. Introduction

1.4 Thesis Organization

Chapter 2 consists of brief description of the related work done on weapon detection. Chapter 3 describes the theory related to object detection. Chapter 4 describes implementation of three proposed methods. Results are presented in Chapter 5. Finally Chapter 6 contains conclusions and scope for the future work.
Chapter 2

Related Work

History of CCTV cameras began in 1942 in Germany. They were deployed to monitor the launch of V2 rocket [11]. In year 1951, first Video Tape Recorder (VTR) was used to capture live images from television camera [12]. In the year 1973, Charge Coupled Device (CCD) technology was invented which made the deployment of surveillance cameras in business sectors possible by 1980 [12]. In 1996, a Swedish company called Axis communication designed worlds first network camera Neteye 200 which was successful in transforming video surveillance from analog to digital [12]. After transformation of analog video surveillance footage to digital, it became possible to apply various image processing, computer vision and machine learning algorithms to recorded videos. In 2003, face recognition algorithms were used in surveillance video for the first time to track missing children by Royal Palm middle school Phoenix [12]. Later, various object recognition algorithms were proposed in the field of image processing and computer vision that made surveillance more and more efficient over time. With the evolution of object detection algorithms, surveillance cameras were being used in various sectors like, access control in spacial areas, person specific identification in certain scenes, crowd flux statics and congestion control, anomaly detection and deterrence, interactive surveillance with the help of multiple cameras [13]. Object detection and tracking with the help of multiple cameras is briefly discussed in [14]. Here the authors explained geometric constraints that are needed to be considered to track an object with the help of multiple cameras. They also discussed few global approaches like 2-D appearance model, 3-D face tracking with geometric face models and few feature based models for object recognition through video cameras.

Authors in [13] discussed methods for detection and tracking of moving objects with motion segmentation algorithms like background subtraction, temporal differencing, optical flow. Object detection with the help of mean shift segmentation was introduced in [15]. Here authors also discussed a method to detect occluded objects with depth information obtained from stereo vision. Detected objects are tracked with Bayesian Kalman filter coupled with simplified Gaussian mixture (BKF-SGM) algorithm. Authors in article [16] proposed methods to evaluate
the performance of different object recognition algorithms. Later in 2005, number plate recognition technique using sliding window mechanism was proposed in [10]. Authors proposed a method that uses sliding window for segmentation and character recognition using neural networks for number plate detection. In [17], Dalal described a novel feature extraction method called histogram oriented gradient (HOG) which gained popularity in applications for human detection. Later, it was used as a feature extraction in many object detection algorithms. In 2002, Devir proposed a method for automated robbery detection depending on actors pose estimation [18]. In [19], authors described the implementation of real-time firearm detection. Here the authors used sliding window mechanism to determine the region of interest and trained a neural network with image intensity pixels as input vector. Output of the neural network resulted in few false alarms (wrongly classified as positives). The false alarms were removed by reclassification of neural network output with region based descriptors.

Many object detection algorithms were proposed in recent years but most of them focuses on detection of large objects like vehicles, humans, number plates etc. Therefore, literature review on object detection algorithms left us with the impression that despite the number of attempts made to detect suspicious and dangerous events monitoring through surveillance system, very few algorithms were proposed in the field of weapon detection. Author in [20] described briefly how close we are to achieve automated surveillance.

2.1 Weapon Detection in Images and Videos

Concept of firearm detection with surveillance cameras were first studied by Iain Darker and his team in 2007 [21]. Later in 2008, same team for the first time tried to implement automated surveillance methods [22]. In 2009, Darker proposed a SIFT based firearm detection algorithm using motion segmentation method of ROI estimation [23]. As SIFT algorithm is prone to false alarms, authors tried to estimate the ROI precisely using motion segmentation rather than applying SIFT detection on whole image. After determining ROI, SIFT algorithm was applied with in the ROI to detect the firearm (pistol). Along with firearms, another category of weapons that can be considered for detection is knives. Research on knife detection is also being carried out in recent years but the number of papers published so far in the field of knife detection is limited. Kmiec in [24] proposed a method to use HOG feature extraction and Support Vector Machine (SVM) classification method of knife detection.

In [25], authors proposed a method to detect knife edges using dominant edge direction. Here the authors assumed that the knife is centred vertically in detection window and found dominant edges by approximating line segments on
extracted edges and the intensities of edges depend on length of the segment. After determining line segments, corresponding orientation of each line segment was found. If orientation of a line segment with respect to vertical axis is greater than certain predefined threshold, they were discarded. Later, authors created feature vector dependent on intensities of dominant edges and straight line segments. SVM was used to classify the image as knife.

Authors in [7] used a novel approach to use Active Appearance Models (AAM) to detect knife in an image rather than to locate an object that is known to exist in the image to be analysed. Here, authors used Harris keypoints [26] as possible candidate set. They rotated the image 15° each time and trained 24 AAM’s to cover all the variations in 360° orientation of image. Authors evaluated their algorithms using 40 knife images with all the images having plane background mostly white background.

A method is proposed in [6] using possibilistic shell clustering algorithms to compute active contour models of knife differs from other papers that the proposed method tried to compute edge direction of knife. This method determines second degree curves in analytical form and knife is detected depending on the angle between two estimated curves. Authors in the article [6] divided manually a training data set into good examples and difficult examples. 30 positive examples and 60 negative examples were manually selected as a test dataset. They were able to achieve an accuracy of 87%.

In [27], authors used fuzzy classification method for knife detection. Here, the authors used one of the feature descriptors i.e edge histogram described in MPEG-7 visual descriptors [28] as feature vector. Edge histogram describes 5 different types of edges: 4 directional and one non directional edges. This method computes only local edges and creates feature vector of length 80 bins [29]. Authors in [27] achieved 84% classification accuracy using manually selected 573 good positive examples and 6164 negative examples. The data set was manually divided in five different combinations having different number of good positive and negative training examples and obtained best classification rate of 84%.

Authors in [2] described two different algorithms for knife and gun detection. Knife detection was done using edge histogram descriptors. Gun detection was carried out on image, first detecting human in the image and then gun in the person’s hands. Authors in [2] proposed an algorithm that uses edge histogram feature extraction and SVM based classification for knife detection. Results of this algorithm are validated using four fold cross-validation with 2627 samples in each. Hence, a total of 10508 images from the dataset containing 12899 images were used. Results were reported in terms of specificity and sensitivity, where 81% sensitivity and 93% specificity were achieved.
So far we could not locate a publication describing an algorithm that can detect both gun and knife. Number of papers proposed on knife detection are less and no papers mentioned above were implemented on real time scenarios. Computation time, which is a major factor when it comes to real time implementation, is not mentioned in any papers. Real time implementation refers to detection of knife in the hands of the person on a complete image of resolution 640 \times 400 \text{ pixels} or higher.
Chapter 3

Object Detection

For object detection algorithms it is important to find a suitable feature vector that would accurately describe the object of interest as that feature vector should be given as input to the neural network for training and testing. For training a neural network to detect an object, a proper input vector should be required. Quality of classification of neural network greatly depends on type of input we are providing to it. Input to the neural network can be direct image pixel intensities or feature vectors that would explicitly describe unique features in required object of interest [28]. Feeding direct image pixel intensities for a neural network works well for applications with constant background and mostly binary images like classification of numbers or alphabets in Optical Character Recognition (OCR) [30]. For applications like weapon detection, human detection through surveillance camera images, backgrounds and shape of objects to be detected would slightly change from one CCTV image to another CCTV image. Therefore, neural network trained with image pixel intensities would have a poor performance.

In this chapter we will discuss theory related to general object detection. As explained in section 1.3 first stage of our object detection algorithm would be segmentation.

3.1 Segmentation

For object detection algorithms like detecting knives or humans as explained in section 1.3.1 we need to calculate local features. Therefore large image need to be divided into small segments. The process of segmentation is carried out through sliding window mechanism.

3.1.1 Sliding Window

Segmentation is carried out linearly and each segmented frame is tested for the presence of object of interest in it. If the object of interest is not found in the segment, the sliding window is moved horizontally by certain number of pixels...
great than one (since moving by 1 pixel does not have any change in image where as moving by larger values would miss the required object of interest). Once the sliding window reaches the last pixel in the row of image matrix, it slides vertically by certain number of pixels and starts investigating again from left most pixel in the image. Once if an object of interest is detected in a segment, then the segment is marked with bounding box containing object of interest using a rectangle to alert the CCTV operator.

3.2 Feature Extraction

Feature extraction is a process of extracting unique features like shape, color, pattern, texture, contour etc. of the object of interest. Features can be of two types, geometric features and appearance features. Geometric features describe size and position of object of interest in the image whereas appearance features describe how appearance of region of interest changes [17]. Various feature detection algorithms like edge detection, Harris keypoint detection (corner detection), Scale Invariant Feature Transform (SIFT), Histogram Oriented Gradient (HOG) etc. were being widely used. For example, authors of the paper [2] used homogeneous texture descriptor and edge histogram containing 62 and 80 elements in feature vectors respectively. Edge histogram describes four directional edges and one non directional edge. Directional edges include edges oriented in $0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$. However classification results obtained by the authors in [2] using homogeneous texture descriptor were not satisfactory. Therefore finding a suitable feature vector is essential for any object detection algorithms. In this thesis HOG feature extraction was found to be suitable for knife and human detection as explained in section 1.3.2.

3.2.1 HOG Feature Extraction

HOG is a powerful feature descriptor primarily used in object detection [17]. It is a feature extraction method developed by Navneet Dalal and Bill Triggs. The main idea of this method is to describe the shape of the object by using intensity gradients or edge directions[17]. In this method the whole image is divided into small blocks or cells and the features of individual cells are extracted. After extracting the features of each cell, cells are grouped together and normalized. By this, features obtained are contrast normalized [17].

HOG feature extraction is carried forward in four stages as shown in figure 3.1.
3.2.1.1 Gradient Computation

Input image for a HOG descriptor can be represented in several colour models like RGB (Red, Green, Blue), LAB (It is a form of colour representation where L for lightness and A and B for the colour-opponent dimensions) and gamma. As most of the surveillance cameras would be RGB cameras we can consider input image to be RGB image which can be later converted into grayscale image for processing. Feature extraction is started from gradient computation. Many discrete derivative masks like 1D point derivatives, diagonal derivatives, cubic corrected, Sobel masks can be used in gradient computation of pixel [17]. For gradient computation the edge directions of image segment $I$ are obtained by convolving sliding window with horizontal ($x_{mask}$) and vertical ($y_{mask}$) masks respectively by placing zero of the mask on the pixel being computed:

$$y_{mask} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \quad x_{mask} = \begin{pmatrix} 1 & 0 & -1 \end{pmatrix}$$

(3.1)

$$\text{Gradient} \quad |G| = \sqrt{I_x^2 + I_y^2}$$

(3.2)

$$\text{Orientation} \quad \theta = \arctan\left(\frac{I_y}{I_x}\right)$$

(3.3)

where values $I_x$ corresponds to horizontal edge directions and $I_y$ corresponds to vertical edge directions.

After finding horizontal and vertical edge directions ($I_x, I_y$), gradients $|G|$ and orientations $\theta$ for all pixels in image $I$ are calculated by equations 3.2 and 3.3 respectively.

3.2.1.2 Orientation Binning

Every pixel contributes a weighted vote depending on the orientation $\theta$ of the gradient $|G|$. For example if computed orientation $\theta$ of a pixel is $15^\circ$, then the weight of bin representing $0^\circ$ to $20^\circ$ increases. These votes are piled up into orientation bins over each cell. These orientation bins can be signed or unsigned. For unsigned gradients, bins are between $0^\circ$-180$^\circ$ and for signed gradients, bins are between $0^\circ$-360$^\circ$. Votes can be any function of the gradient magnitude or the magnitude itself [17].
3.2.1.3 Descriptor Blocks

Descriptor blocks can be divided into two types, rectangular blocks (R-HOG) and circular blocks (C-HOG). These blocks are combination of cells as shown in figure 3.2. Block sizes are usually $2 \times 2$ or $4 \times 4$ cells for rectangular blocks as shown in figure 3.2(a). Circular blocks might contain single central cell or along with central cell it may also contain angularly divided cells as shown in figure 3.2(b). Normalization can be done on a block in order to restrict variations in computed histogram bins due to variations in illuminance of pixels. Usually rectangular blocks are preferred as they are easy to implement.

![Figure 3.2: Rectangular and circular block descriptors.](image)

3.2.1.4 Block Normalization

To make HOG method invariant to illuminance, local contrast normalization is essential [17]. Block normalization can be done in four different ways. Normalized feature vector $f$ is given by following equations,

\[
L2\text{-norm} : f = \frac{v}{\sqrt{||v||^2_2 + e^2}} \quad (3.4)
\]

\[
L1\text{-norm} : f = \frac{v}{||v||_1 + e} \quad (3.5)
\]

\[
L1\text{-sqrt} : f = \frac{v}{\sqrt{||v||_1 + e}} \quad (3.6)
\]

where $e$ is a very small random value between 0 and 1 [17], $v$ is the vector consisting of non-normalized histogram bins in each block and $||v||_k$ is the $k$-norm for $k=1,2,...$
3.3 Classification with Neural Networks

After extraction of feature vector neural networks are used for classification. Neural networks theory is motivated by neural networks in human brain which performs complex computations in no time [31]. Human brain organizes its components called neurons in certain way to perform many computations like pattern recognition or perception much faster than modern computers [31]. The ability of a neural network is to learn from its environment [31]. This learning is carried out in iterations. For each iteration, the neural network becomes more knowledgeable [31]. There are different kinds of neural networks [31]. For this thesis multiple feed forward neural network is used.

3.3.0.1 Multiple Feed Forward Network

A feed-forward network is a neural network in which connections between nodes do not form loops or cycles [31]. They propagate information in a single direction. Figure 3.3 shows the implementation of feed forward neural network. Usually neural networks have an input layer, an output layer and one or more hidden layers. If the number of hidden layers in a neural network is more than one, then the neural network is called a multilayer neural network [31]. If the number of hidden layers are greater than two, then the neural network is a deep neural network [30]. The performance of a neural network increases as the number of hidden layers increase and the performance varies very slightly after certain number of hidden layers [30]. However complexity of neural network increases as the number of hidden layers increase [31].

![Neural network diagram](image)

Figure 3.3: Neural network.
3.3.1 Back-Propagation algorithm

As number of hidden layers increase, a large number of complex functions can be learned [31]. However, the complexity increases to calculate the correct combination of weights [31]. For handling such multilayer networks, back-propagation method is used [31]. In back-propagation learning procedure a randomly selected input vector from training set is fed as input to the neural network with some randomly initialized weights and the network produces an output vector that is to be compared with target vector [32]. Weights of the network are updated by using the difference between the output vector and desired target vector. The process of weight correction is continued until the difference between target vector and output vector becomes minimum. This process of error correction is carried out for many iterations. The gradient of the error function is calculated at each iteration. The activation function selected should be continuous and differentiable [32]. If $x$ is the input vector, $w$ are the weights of the hidden layers and $y$ is the output vector then

$$y_j = \sum x_i w_j$$  \hspace{1cm} (3.7)

where $y_j$ is the $j^{th}$ layer output for the input $x_i$, $i$ will be $j - 1$.

3.3.1.1 Activation Function

There are two kinds of activation function used in backpropagation algorithm [31], unipolar sigmoid function and bipolar sigmoid function. A unipolar sigmoid function is defined as

$$Y_j = \frac{1}{1 + e^{-x_i w_j^T}}$$  \hspace{1cm} (3.8)

Bipolar sigmoid function is given by

$$Y_j = \frac{1 - e^{-x_i w_j^T}}{1 + e^{-x_i w_j^T}}$$  \hspace{1cm} (3.9)

3.3.1.2 Training a Neural Network Using Back-Propagation Algorithm

For training a neural network to detect an object, large number of positive and negative samples are used. Positive samples refer to sample images that contain required object of interest whereas negative samples do not. Though there can be any number of layers in the neural networks for limiting the computational cost we are considering four layer neural network in this thesis. In four layer neural network, let $N$ be the size of input vector, first hidden layer consists of $n1$ number of neurons, second hidden layer consists $n2$ neurons and output layer consists of $n3$ neurons. Four layer neural network contains 3 weights $w1, w2, w3$. Neural network is initialised with random weights, $w1$ containing $N \times n1$ weights, $w2$
containing $n_1 \times n_2$ weights and $w_3$ containing $n_2 \times n_3$ weights. Implementation model of four layer neural network is shown in figure 3.3.

Weights of each layer are initialized using following formula.

$$w_1 = \frac{\text{rand}(N,n_1)-0.5}{N}$$  \hspace{1cm} (3.10)  
\[w_2 = \frac{\text{rand}(n_1,n_2)-0.5}{n_1}\]  \hspace{1cm} (3.11)  
\[w_3 = \frac{\text{rand}(n_2,n_3)-0.5}{n_2}\]  \hspace{1cm} (3.12)

where MATLAB command \text{rand}() produces a vector containing values randomly distributed in the range $[0,1]$.

After initialization of the weights, input-target pair is considered randomly and network output $y$ is computed. After computation of output $y$, error $e$ between target $t$ and output is calculated according to equation 3.13.

$$e = t - y$$  \hspace{1cm} (3.13)

To correct the weights using back-propagation algorithm, local error $\delta_i$ is computed for each layer $i$. $\delta_i$ for output layer is calculated using following equations [31]:

$$\delta_3 = \text{diag}(e) \cdot \frac{\partial y}{\partial w_3}$$  \hspace{1cm} (3.14)

where \text{diag}(e) is the diagonal matrix representing the error vector, bipolar sigmoid function given by equation 3.9 is used as an activation function for the neural network. Hence equation 3.14 becomes

$$\delta_3 = \text{diag}(e) \cdot (1 - \tanh(y)^2).$$  \hspace{1cm} (3.15)

Correction in weights $\Delta w_3$ is given by

$$\Delta w_3 = \alpha \cdot \delta_3 \cdot O_2$$  \hspace{1cm} (3.16)

where $\alpha$ is the learning rate and $O_2$ is the output vector obtained from the second hidden layer. Similarly $\delta_2$ and $\Delta w_2$ are calculated by

$$\delta_2 = ((w_3)^T \cdot \delta_3)(1 - \tanh(O_2)^2)$$  \hspace{1cm} (3.17)  
$$\Delta w_2 = \alpha \cdot \delta_2 \cdot O_1$$  \hspace{1cm} (3.18)

where $O_1$ is the output of first hidden layer and $\delta_1$ and $\Delta w_1$ are given by

$$\delta_1 = ((w_2)^T \cdot \delta_2)(1 - \tanh(O_1)^2)$$  \hspace{1cm} (3.19)  
$$\Delta w_1 = \alpha \cdot \delta_1 \cdot x$$  \hspace{1cm} (3.20)
where $X$ is the input vector. With the help of correction factors $\Delta w1$, $\Delta w2$, $\Delta w3$ new weights can be calculated using equations 3.21.

$$
\begin{align*}
    w3 &= w1 + \Delta w3 \\
    w2 &= w2 + \Delta w2 \\
    w1 &= w1 + \Delta w1
\end{align*}
$$

(3.21)

With help of the new weights, error vector $e$ is again calculated considering another example from the training set. The process of weight correction is repeated until mean square error becomes constant.

### 3.3.2 Ten-Fold Stratified Cross-Validation

To test the performance of the neural network accurately, it is necessary to use every sample in the database as a training example and testing example. This process of testing is done by using 10 fold cross-validation [33]. In 10 fold cross validation, available data is divided into 10 equal sub-sets of which 9 sets are used as training sets and $10^{th}$ set is used as test set [33]. In the next iteration, one set is replaced from the training set for test and previous test set is included in the set of training subsets. This process is continued till all the available sets are used as both test set and training set. In 10 fold stratified cross-validation, positive and negative samples in each set for training and test are divided in equal proportions.

### 3.3.3 Performance Evaluation of a Classifier

Neural network is designed using back-propagation as explained in section 3.3.1.2. Performance of a neural network depends on various factors such as number of training samples, number of neurons in hidden layers, number of iterations and learning rate $\alpha$. Therefore to design an optimally performing neural network we need to create different neural networks varying different parameters mentioned above and a Receiver Operating Characteristics (ROC) graph should be plotted for each designed network. Author in [34] explained ROC graph as, "An ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance ". ROC graph is the plot of correctly classified positive samples vs falsely classified positive samples. From this ROC graph we need to select the neural network that produces low false positives and high true positives.

When a trained classifier is tested on test dataset, there are four possible outcomes [34]. If the instance is positive and output of the classifier is positive then outcome is referred as true positive (TP). If the instance is positive and classifier output is negative then outcome is referred as false negative (FN). If the instance is negative and output is negative then outcome is referred as true
negative (TN) and finally if the instance is negative and output of the classifier is positive then outcome is referred as false positive (FP). From the four outcomes confusion matrix can be drawn [34].

<table>
<thead>
<tr>
<th>TP</th>
<th>FP</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>FP</td>
<td>PPV 1-PPV</td>
</tr>
<tr>
<td>FN</td>
<td>TN</td>
<td>FOR 1-FOR</td>
</tr>
<tr>
<td>TPR</td>
<td>TNR</td>
<td>Accuracy 1-Accuracy</td>
</tr>
</tbody>
</table>

Table 3.1: Confusion matrix

\[
\text{True Negative rate} = \frac{\text{True Negatives}}{\text{Total Negatives}} \tag{3.22}
\]

where total negatives is given by the sum of false positives and true negatives

\[
\text{True positive rate} = \frac{\text{True Positives}}{\text{Total Positives}} \tag{3.23}
\]

where total positives is given by the sum of true positives and false negatives.

\[
\text{Positive Predictive Value (PPV)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.24}
\]

\[
\text{False Omission Rate (FOR)} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Negatives}} \tag{3.25}
\]

\[
\text{Recall (sensitivity)} = \frac{\text{True Positives}}{\text{Total Positives}} \tag{3.26}
\]

\[
\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Positives} + \text{Total Negatives}} \tag{3.27}
\]

\[
\text{Specificity} = \frac{\text{True Negatives}}{\text{False Positives} + \text{True Negatives}} \tag{3.28}
\]
3.4 Neural Network for Classification of Single Class

For classifying the image as an image containing object of interest or not, HOG feature vector calculated for that image should be used as an input vector $x$ to the neural network. To detect single type of object one can use a neural network with one neuron in the output layer which generates a value one when there is an object of interest in the image and zero in the absence of object of interest. For training and testing single class classifier, training and test dataset should contains thousands of positive and negative examples.

3.5 Neural Network for Classification of Multiple Classes

To detect multiple kinds of objects we need more than one neuron in the output layer. Multi class classifier was widely used in Optical Character recognition (OCR) system to classify different characters in the a text. In this thesis similar to OCR system a four layer neural network with three output neurons in the output layer, a multi class classifier was designed to detect gun and knife. A multi class classifier is a network which is capable of classifying multiple types of images. Therefore multi class classifier should contain more than one neuron in the output layer. If each neuron of output layer represents one class then the neuron that outputs the highest value is considered as the output class determined by the network. For training these kind of neural networks, training and test data set should contain few hundreds to thousands of samples representing each class.

3.6 Image Databases for Evaluation of Performance of Different Methods

In this thesis three kinds of neural networks were designed. One is for knife detection in an image, second one is for detecting two kinds of weapons i.e is knives and guns in an image and the third network is to detect humans in an image. Therefore to train these neural networks different datasets containing positive and negative samples were required.

3.6.1 Image Database of Knife

For training a neural network using back-propagation algorithm for knife detection, a database of positive and negative sample images are required. Positive samples which contains images of knife in the hands of humans and negative samples i.e image without knives were created by authors of [2]. Same database is used for training a neural network presented in this report. Positive samples
Chapter 3. Object Detection

contain only images of knife held in the hand. As knife in an image without hand is not considered as threat, those kind of images were not included in the positive samples. Negative samples contain random images taken in similar environment without knife. The database consists of 12899 images of which 3559 are positive samples and 9340 are negative samples. For efficient neural network training, it is required to have more number of negative samples compared to positive samples [2]. Examples of positive and negative samples for knife detection are shown in figures 3.4(a) and 3.4(b) respectively.
Chapter 3. Object Detection

(a) Positive examples showing knife.

(b) Negative Examples Without Knife.

Figure 3.4: Positive and negative examples
3.6.2 Image Database of Gun

For training a neural network to detect both gun and knife, positive samples should also contain images of gun. As there is no public database with images of gun, a nominal database containing 309 training and 349 test images were collected for this work. All these images were downloaded from Google images. Figure 3.5 shows the examples of training and testing samples used. Table 3.2 describes the number of images used for training and testing the classifier.

![Train Samples](image1)

![Test Samples](image2)

Figure 3.5: Training and testing samples for multi-class classifier.

<table>
<thead>
<tr>
<th>Type</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive samples (knife)</td>
<td>1392</td>
<td>2104</td>
</tr>
<tr>
<td>Positive samples (guns)</td>
<td>309</td>
<td>341</td>
</tr>
<tr>
<td>Negative samples</td>
<td>6079</td>
<td>3234</td>
</tr>
</tbody>
</table>

Table 3.2: Dataset division for multi-class classifier

As all the images have white background and are not in the hands of human neural network trained using these images would not perform well for real time images that is detecting a gun in the hands of the person. Hence these neural
networks were not tested on our real time $640 \times 480$ pixels images.

### 3.6.3 Image Database of Human

To train the neural network to detect humans INRIA database is used [17]. It contains 2416 positive training samples of $128 \times 64$ pixels resolution and negative training samples contains 1218 images with different resolutions larger than $128 \times 64$ pixels. From each of these 1218 negative sample images, five images of resolution $128 \times 64$ pixels is cropped randomly and produced a training set containing 6090 images of negative samples.

![Positive examples for human detection.](image-url)
Test set contains 1126 positive samples and 453 images which are negative samples of higher resolution. 3 images are cropped randomly from each of these 453 images to create a test set containing 1359 test images with negative samples. Table 3.3 shows the number of training and testing samples used for the human detection.

Table 3.3: Positive and negative training sets for human detection

<table>
<thead>
<tr>
<th>Type</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive samples</td>
<td>2416</td>
<td>1126</td>
</tr>
<tr>
<td>Negative samples</td>
<td>6090</td>
<td>1359</td>
</tr>
<tr>
<td>Total samples</td>
<td>8506</td>
<td>2485</td>
</tr>
</tbody>
</table>
Chapter 4

Implementation

A CCTV image with minimum acceptable resolution of $640 \times 480$ pixels is considered in this work, as main aim of the thesis is to detect a knife in low resolution image. Implementation of the project was done in MATLAB R2016a and test images of person carrying knife are taken with the mobile camera with a resolution setting of $640 \times 480$ pixels. Compared to size of total image, size of knife could be very small and therefore knife is considered to be present in a window of size not more than $100 \times 100$ pixels. Another consideration for assuming that knife fits in an image of size $100 \times 100$ pixels is as CCTV cameras can be installed at a rather far place from human i.e to ceiling.

4.1 Knife Detection in the Original Image

Overview of proposed solution of the problem of knife detection through surveillance camera images was provided in a flowchart in figure 1.1, see method 1 block. The proposed solution consists of various steps which are discussed in following sections.

4.1.1 Segmentation by Using Sliding Window

For the process of segmentation a $100 \times 100$ pixels sliding window was used. For knife detection in the image of resolution $640 \times 480$ pixels image this sliding window was started from top leftmost pixel and each segmented image was tested for the presence of knife. After classification of the segment the sliding window was moved horizontally by 20 pixels as explained in section 3.1.1 till the end in the row of image matrix was reached. Then the sliding window was shifted 20 pixels in vertical direction and started scanning from the left most pixel. This process was continued till the end of the image was reached.

4.1.2 HOG Based Feature Extraction for Knife

After extraction of $100 \times 100$ pixel segments through sliding window mechanism one after another, corresponding HOG features were calculated for each segment.
Horizontal and vertical derivatives for the segmented images were calculated as explained in section 3.2.1.1. Then the $100 \times 100$ pixels segments were divided into cells of $8 \times 8$ pixels containing 64 pixels in each cell. Unsigned orientation with 9 histogram bins was used. Therefore for each cell 9 bin histogram representing orientations ranging from $0^\circ$-$180^\circ$ was computed. Rectangular blocks were formed using 4 cells each with 50% overlapping and cells in the block were normalized using L2-Norm given by equation 3.4, to produce 36 bins per block. For $100 \times 100$ pixel image, a total of 121 ($11 \times 11$ where 11 are the number of blocks in each row and each column) overlapping blocks were formed. Therefore a feature vector of length 4356 bins was obtained.

### 4.1.3 Neural Network for Classification of Single Class (Knife)

As explained in section 3.3.3 four classifiers were designed by varying the number of training examples and the number of neurons in hidden layer to determine the possible combination of the network with capability to detect the knife with highest accuracy. Parameters of these combinations were as follows:

1. **Classifier A:**
   - (a) First hidden layer: 25 neurons
   - (b) Second hidden layer: 25 neurons
   - (c) positive training examples: 1392
   - (d) Negative training examples: 6079

2. **Classifier B:**
   - (a) First hidden layer: 50 neurons
   - (b) Second hidden layer: 30 neurons
   - (c) positive training examples: 1392
   - (d) Negative training examples: 6079

3. **Classifier C:**
   - (a) First hidden layer: 50 neurons
   - (b) Second hidden layer: 30 neurons
   - (c) positive training examples: 270
   - (d) Negative training examples: 2700

4. **Classifier D:**
   - (a) First hidden layer: 50 neurons
(b) Second hidden layer: 30 neurons
(c) positive training examples: 50 images of knife with plane background
(d) Negative training examples: 50

With the increase in number of neurons in each layer, complexity and computational time for testing and training the network increases but the performance of network does not change after a certain level. If the number of neurons used were very low, neural network would not be capable to classify images properly. Therefore, an optimum number is selected by experimentation.

Different combinations of neural network with different number of neurons in the hidden layer were used to train the neural network. All the neural networks were trained with the learning rate of 0.01 and bipolar sigmoid function was used as an activation function.

The classifier B has 80 neurons in the hidden layers as compared to the classifier A with 50 neurons in hidden layers. Due to greater amount of neurons in hidden layers the classifier B has correctly classified greater number of images as compared to the number of images correctly classified by the classifier A. If the number of neurons in the hidden layer are further increased, computation time increases while the performance of neural network would not increase. In classifier C, number of training samples are reduced to 2970 where as 7471 training samples are used in classifier B. By reducing the size of training set, false positive rate increased. In classifier D knife with only plane background images were considered by which performance level decreased drastically due to low training and the neural network could not classify images with background.

An ROC graph was drawn for all the classifiers as shown in figure 4.1. Dotted line passing through centre of the graph is a random guess line i.e classifier outputs 50 times wrong in every 100 images which means it was a random guess. Classifier B was selected for knife detection application as it has lowest false positive rate and highest true positive rate.
Therefore classifier B parameters were used to design and train the neural network. The trained neural network weights were stored as a new system and tested on new data. For testing the accuracy of neural network, feature vectors were calculated from 5365 new images that were not used for training and the neural network with new weights was tested on the new feature vectors. Among these 5365 new images, 2104 images contain knife and 3261 images were negative samples. Database contains 3559 positive examples but only 3496 examples are used. Knife in the remaining examples was not visible hence these images were not considered.

Table 4.1: Positive and negative training sets for knife detection

<table>
<thead>
<tr>
<th>Type</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive samples</td>
<td>1392</td>
<td>2104</td>
</tr>
<tr>
<td>Negative samples</td>
<td>6079</td>
<td>3261</td>
</tr>
</tbody>
</table>

4.1.4 Detection of Knife in a Real Time Image

Classifier B was used to detect a knife in an whole image taken by a CCTV camera. The neural network produced an output "1" when knife was present in
the image and a bounding box was drawn to alert the CCTV operator. In this thesis the neural network was trained such that if the output of the activation function of output neuron was greater than threshold 0.8, output of the function was marked as knife and a bounding box was drawn surrounding knife. If the value was less than threshold then the considered segment was marked as safe and discarded. Threshold was chosen as 0.8 because neural network was trained to output a value close to 1 when knife was present and close to 0 when the output was negative sample.

In method 1 as explained in flow chart in figure 1.1 the designed classifier is trained to detect knife in the image a prior human detection. Therefore the proposed algorithm detected the knife in the image to alert the CCTV operator even if the knife was not in the hands of a person (not a threat) as shown in the figure 4.2(a).

Figure 4.2: Real life example images

Figure 4.2(b) shows the image of a person carrying the knife in the hand which is considered to be a threat. Hence in other 2 methods we will find the human in an image and detect the knife if a human was detected.

4.2 Human Detection Before Knife Detection

As knife in an image without human would not be considered as a dangerous situation, before finding a knife it would be efficient to find a human in an image. If human was found successfully in an image then proposed algorithm could find the knife in the surroundings of human region while discarding the rest of the image. For the purpose of human detection, two methods were used by us. As
explained in flowchart in figure 1.1. Method 2 is to find human through well
known HOG algorithm coupled with neural networks. Method 3 is based on the
idea that for CCTV surveillance, background can be considered constant in some
cases. Therefore an image frame containing a background and a human can be
subtracted from the image frame containing only background. Consequently a
knife can be detected in the image containing the result of such subtraction.

4.2.1 Human Detection Using HOG Method

4.2.1.1 Segmentation by Sliding Window

In this method, the detection of human in the image was followed by search for a
knife in the hands of human. To detect a human we divided our $640 \times 480$ pixels
image into segments of $400 \times 200$ pixels each using sliding window technique. Since
CCTV cameras can be installed at a rather far place from human, for our thesis
it was assumed that entire image of a human fits into a $400 \times 200$ pixels segment
of image. Segmentation was carried out according to sliding window technique
described in section 3.1.1. To detect human in an image a horizontal and verti-
cal shifts of 20 and 50 pixels correspondingly were used to move the sliding
window to next position. After the segment with a human was obtained, image
area around the human was considered and segmented into smaller segments of
$100 \times 100$ pixels each. Then a smaller segment was verified for the presence of
knife in it. This segmentation was carried out by sliding $100 \times 100$ pixels window
as explained in section 3.1.1 until a knife was found in a frame. If no knives are
detected, previous step was repeated to detect other human by segmenting with
sliding window of size $400 \times 200$ pixels. This process was repeated until the whole
image was processed.

4.2.1.2 HOG Feature Extraction

After segmenting the $400 \times 200$ pixels image segments from $640 \times 480$ pixels
image one after the other, features for each segment were extracted using HOG
feature descriptor as explained in section 3.2.1. Before extracting features, to
reduce the number of computations, image segment was down-sampled to $128 \times 64$
pixels. Horizontal and vertical edge directions, gradient and orientations were
computed for all the pixels as explained in section 3.2.1. Figure 4.3(b) and (c)
shows the example of the computed vertical and horizontal edges for the image
4.3(a) respectively.
Chapter 4. Implementation

Figure 4.3: Horizontal and vertical edges of human.

Now $128 \times 64$ pixels image segments were divided into cells of $8 \times 8$ pixels containing 64 pixels in each cell. Unsigned orientation with 9 histogram bins for each cell were used for orientation binning. Each pixel voted certain weight for the orientation on the basis of gradient magnitude calculated for that particular pixel from equation 3.2. After binning according to orientation, rectangular blocks of 4 cells were formed with 50% overlapping. Cells in this block were normalized using L2-Norm to produce 9 histogram bins for each cell x 4 cells per block = 36 normalized bins were computed for each block. For $128 \times 64$ pixel image segment, a total of 105 ($15 \times 7$ where 15 are the total number of blocks in each column and 7 are the total number of blocks in each row) overlapping blocks were formed. Therefore a feature vector of length (105 rectangular blocks with 36 bins in each block) 3780 bins was obtained.

The extracted HOG feature vector was given as input to neural network to detect the humans. If a human was detected then knife detection algorithm would proceed to search for knife surrounding human region.

4.2.1.3 Neural Network for Human Detection

A neural network was trained to detect humans in the segmented images. Features calculated in section 4.2.1.2 were fed as input to the neural network to test the presence of human in the image segment. A 4 layer neural network with 50 and 30 neurons in the corresponding hidden layers was designed to classify human in an image. Input layer contains 3780 neurons as size of the HOG feature vector obtained in section 4.2.1.2 was 3780 bins. Neural network was trained with bipolar sigmoid function and learning rate of 0.01 as explained in section 4.1.3. It took 150 iterations for the MSE to converge. Examples of positive and negative samples used for training and testing were shown in figures 3.6 and 3.7 respectively.
Chapter 4. Implementation

Trained and tested classifier was used to detect the humans in the image acquired in our work. To detect the human, feature vectors computed for each segment as explained in section 4.2.1.2 were used as input to the neural network for classification. Threshold for activation function was set to 0.8 as bipolar sigmoid threshold outputs a value close to 1 if the output was a positive sample. When the output of neural network was greater than 0.8, image segment was classified as positive and when the output was less than 0.8, image segment was classified as negative sample.

4.2.2 Human Detection Using Background Subtraction

When surveillance was used indoors like banks, offices, schools and shopping malls, the background can be considered as constant. Hence ROI around the humans can be selected by subtracting reference background frame from the considered image frame. Algorithm for subtracting background frame is shown as flowchart in figure 4.4.

![Flow chart of background subtraction](image)

Figure 4.4: Flow chart of background subtraction.

As shown in the flow chart, extracted surveillance video image frame was compared with background image frame and absolute difference (R) was calculated between image frame and background according to equation 4.1.

\[ R = |I_r - I_b| \]  \hspace{1cm} (4.1)

where \( I_r \) were the pixel intensities of image under consideration and \( I_b \) were the pixel intensities of reference background frame. Then the difference R was compared with predefined threshold. Here the threshold was considered to be 20 to ignore small changes in the pixel intensities as it was found empirically in this work that two images taken at same time have small variations in pixel intensities. If the difference was greater than predefined threshold, pixel was considered as foreground and if the difference was less than threshold, pixel was considered as background and was discarded.

After subtraction, edges in the image were determined using canny edge detection algorithm [35]. The edge image presumably contains edges at the positions
correlated with the position of a human in the original image. Hence we considered a sum of the pixel intensity values inside of a 100 x 100 pixels sliding segment in the edge image as an indicator of possible presence of knife in this segment. For a segment in the edge image where this sum is greater than 100 the corresponding segment in the original RGB image was located using the pixels coordinates of the segment in the edge image. After the segmentation process features were extracted and classified as knife or non knife segments as explained in section 4.1.

4.2.3 Knife Detection Surrounding Detected Human Region

During the process of human detection sometimes knife may be present outside the detected human segment as shown in figure 4.5. Hence, if a knife is searched within the detected human segment then knife might not be found. To compensate this situation, some of the region around the human was also considered for knife detection. If a human was detected in an image then the image segment along with at least 50 pixels surrounding the detected image segment was also considered. Then knife was searched in the image segment by again dividing the image using sliding window into 100 x 100 pixels. For each of these segments HOG features were extracted as explained in section 4.1.2 and used as input to the neural network trained to detect knife as explained in section 3.3.1.2. From the output of the neural network we would draw a bounding box if a knife was found in the image.

Figure 4.5: Example showing knife outside detected image segment.
4.3 Knife Detection in Recorded Video

Implementation of the proposed algorithm was done on recorded video. Video was recorded by us with the help of mobile camera with $640 \times 480$ pixels resolution to test the performance of the algorithm on videos. As background subtraction method was found to be fastest among the three methods this method was used in detection of knife in the video. Two videos with different background were created and tested. Videos were recorded with 30 frames per second frame rate and both videos were approximately 25 seconds each. Testing was done by extracting the frames from the video and subtracting the frame under consideration with the reference background. After detecting the knife in each frame it was observed that some of the false positives appeared at different positions in each frame. Hence to reduce the number of false positives 4 consecutive frames were compared with each other and the detected segments with common coordinates were marked as positives. By this method we were able to reduce the number of false positives but we were not able to discard all false positives.

4.4 Detection of Knife and Pistol

4.4.1 Neural Network as Multi-Class Classifier

As explained in chapter 2 few algorithms were proposed to detect knife and two algorithm [2, 23] to detect gun in an image. In this thesis a novel approach was initiated to design a classifier that is capable to detect two kinds of weapons, i.e. knife and gun (only pistol). Later it can be extended to different types of guns. For the purpose of classifying multiple types of weapons in an image a multi class classifier need to be designed similar to Optical Character Recognition (OCR) system [30].

4.4.2 Training a Multi-class Classifier

To detect knife and gun in an image using single neural network a four layer multi class classifier was trained with the configuration similar to single class classifier. However here we used three neurons in the output layer. From ROC graph of a single class classifier as it was found that classifier B has low false positive rate and high true positive rate for single class classification, the same classifier configuration with three neurons in the output layer was used to design a multi class classifier. It took 150 iterations for the convergence of MSE when 0.01 was chosen as learning rate. First neuron was activated when knife was detected, neuron 2 was activated when gun was detected and neuron 3 was activated when neither knife nor gun was detected. For training a neural network, 1392 images of knives and 309 images of guns were used as positive samples and 6079 examples of neither knife nor gun were used as negative samples. Target pair for knife was
considered as [1 0 0] and for gun was [0 1 0]. Target pair in the absence of both knife and gun was [0 0 1].

For real time implementation of this algorithm a new database containing images of guns held in the hands of the person need to be created. Therefore neural network with different configurations such as different length of training sets, different number of neurons was not designed and evaluated to plot an ROC graph to select the best classifier.

### 4.4.3 Parameters for Neural Networks

Overview of the parameters used for the design of three types of neural networks are tabulated in the table 4.2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Knife</th>
<th>Human</th>
<th>Knife and gun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of layers</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Number of neurons in input layer</td>
<td>4356</td>
<td>3780</td>
<td>4356</td>
</tr>
<tr>
<td>Number of neurons in 1st hidden layer</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Number of neurons in 2nd hidden layer</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Number of neurons in output layer</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>120</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

From the table we can observe that all the neural networks used in this thesis were four layer neural networks called deep neural networks [30]. As number of neurons in the input layer depends on the feature vectors used to describe the image, length of input vectors was different for weapon detection and human detection in this work.
Chapter 5

Results and Discussion

5.1 Detection of Knife in Image

A 100 × 100 pixels sliding window was used to extract segments from the 640 × 480 pixels image and corresponding feature vector was calculated for each segment as explained in section 4.1.2. Example of calculated HOG features for knife was shown in figure 5.1. In figure 5.1, white streaks show the computed HOG features of the image segment.

![HOG feature descriptor for knife.](image)

Figure 5.1: HOG feature descriptor for knife.

The calculated feature vectors were used as input to the classifier B one after the other. During the training of the neural network, amount of correct classification was calculated by computing Mean Square Error (MSE) of the network after each iteration. MSE is the sum of the squares of errors obtained for each input divided by number of inputs considered. From the MSE graph in figure 5.2 we can see that amount of error after every iteration was reduced.
Figure 5.2: Decrease of MSE with increase in number of iterations for neural network training for knife detection.

Trained neural network was tested on the test data set presented in table 4.1. Figure 5.3 shows the confusion matrix of single class classifier B on test dataset. TPR, TNR, PPV, FOR and accuracy were calculated according to the equations 3.23, 3.22, 3.24, 3.25, 3.27 respectively. Columns represent target classes of images of which, column 1 represents knives and column 2 represents images with negative samples. Rows represent output class determined by the neural network. Row 1 represents knife and row 2 represent negative samples. Green boxes represent correctly classified images and orange boxes represent wrongly classified images. From confusion matrix accuracy of classifier B was found to be 84.6%.

Figure 5.3: Confusion matrix for single class classifier
This trained and tested classifier B was used to classify the knife in our 640 × 480 pixels image. Results obtained from the neural network were shown in figure 5.4.

In figure 5.4, rectangular boxes represent segments which were classified as positives by the neural network. By implementing detection on the whole image, there might be more number of false positives as number of segments need to be classified is greater as compared to a number of segments to be classified in a part of the image. Therefore, human in an image was detected before the detection of knife and it was done using two methods: background subtraction method and HOG feature extraction method for human detection.

5.2 Background Subtraction Method

5.2.1 Background Subtraction

As CCTV camera considered as fixed, initially a background image was captured after the installation of camera and used as a reference background. Example of a reference background and a human carrying weapon in indoor environment is shown in figure 5.5. From figure 5.5(a) we can see the background image and figure 5.5(b) shows the image of the person carrying the weapon with same background.
The result of background subtraction is shown in the figure 5.6. Figure 5.6(a) shows background subtracted image which contains only human. Figure 5.6(b) shows the edges in background subtracted image.

Figure 5.6: Background subtraction and canny edge detection output.
5.2.2 Knife Detection After Background Subtraction

After subtraction of background from the image, a $100 \times 100$ pixels sliding window was used to detect knife in the original image by ignoring the background segments. Features of each segment in original image at corresponding positions where sum of the pixels in the canny edge detected image segment greater than 100 were calculated as explained in section 4.2.2. Extracted feature vector was used as an input to the neural network that was trained to detect knife and the output of neural network is shown in figure 5.7. In figure 5.7, a total of 138 $100 \times 100$ pixel image segments obtained after background subtraction were classified by the neural network, of which 3 segments were classified as segments containing knives. Black rectangular boxes in figure 5.7 represent the segments classified as positive by the neural network trained to detect knife.

Figure 5.7: The segments classified as "knife containing segment" by the classifier B are shown by three rectangular boxes.
5.3 Detection of Human Using HOG and Neural Network

5.3.1 Segmentation

As explained in section 4.2.1.1, using sliding window mechanism image segments of size $400 \times 200$ pixels were segmented as shown in figure 5.8. Figure 5.8(a) shows the person carrying a weapon and rectangular boxes shows how sliding window is moved from one segment to another. Figure 5.8(b) and 5.8(c) shows examples of segmented images through sliding window.

![Sliding Window Example](image)

(a)

(b)

(c)

Figure 5.8: Segmentation using sliding window

5.3.2 HOG Representation of Human

HOG features of a image segments were calculated as explained in section 4.2.1.2. Output of the extracted features is shown in figure 5.9, where white streaks represent the computed HOG features of segmented images.
5.3.3 Neural Network for Human Detection

Neural network to detect human in an image was trained as explained in section 4.2.1.3. After training a neural network it was tested on the test dataset mentioned in table 3.3 and a confusion matrix was calculated as explained in section 3.3.3.

![Confusion Matrix](image)

Figure 5.10: Results for INRIA database.

In the confusion matrix shown in figure 5.10, green boxes represent correctly classified outputs. First column represents positive samples i.e. humans and second column shows negative samples. Orange boxes shows the misclassified samples in each class. Results in blue box represent the total accuracy of the classifier.

Extracted features are used as an input to the neural network (trained for human detection) and if a human is detected in the segment then the segment is marked with the bounding box. In figure 5.11 shows the image taken by us and rectangular box shows the detected human.
5.3.4 Knife Detection After Detection of Human

After detection of human the knife is searched in the segment containing human. A rectangular box protruding out in figure 5.12 represents the $100 \times 100$ pixels image segmented by sliding window mechanism for knife detection.

```
Figure 5.12: Sliding window for knife detection.
```

Extracted HOG feature vector for each image segment was used as input to the neural network. Final output of the neural network (trained to detect knife) after classification is shown in figure 5.13. From figure 5.13, a total of 240 image segments are classified by neural network. Among 240 image segments, 4 segments were classified as positive of which 3 were falsely classified. In figure 5.13 black rectangular boxes represent the segments classified as positives by the neural network (trained to detect knife).
5.4 Performance Evaluation of Multi-Class Classifier

Multi class classifier is used to detect multiple types of objects. In this thesis multi class classifier was used for detecting knife and gun. As database for gun was created from Google images which contain images of guns with white background, they were not implemented and tested on our real time 640 × 480 pixels image. In database as shown in table 3.2, 5679 images were used to test multi-class classifier. It contains 2104 images of knives, 341 images of guns and 3234 images of neither guns nor knives. A confusion matrix for this test set was calculated as explained in section 3.3.3. Figure 5.14 shows the calculated confusion matrix for multi class classifier.

![Confusion Matrix](image)

**Figure 5.14:** Confusion matrix of multi-class classifier.
In the figure 5.14 each column represents the target class and row represents the output class. First three columns represent classes and last column represents amount of misclassification and accuracy. First column represents knife class, second column represents gun and third column represents neither knife nor gun. Green boxes represent number of correctly classified objects and orange column represents number of misclassified objects. Blue box represents the accuracy of the classifier.

5.5 Performance Evaluation of Neural Network Used for Knife Detection Using Cross Validation

As explained in section 3.3.2, 10 fold stratified cross-validation was used for evaluating the performance of the neural network on dataset of 12830 images. Sensitivity, specificity and accuracy for each fold were calculated according to the equations 3.26, 3.28, 3.27 respectively and corresponding averages of these values for 10 folds were computed, see table 5.1.

Table 5.1: Results of stratified 10 fold cross-validation

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80.4%</td>
<td>85.70%</td>
<td>83.05%</td>
</tr>
</tbody>
</table>

5.6 Calculation of Computational Time

Computational time required for knife detection methods was calculated in MATLAB. Time required for detection of knife in an image through method 1 i.e sliding window segmentation is 0.93 sec. It was required to scan 513 image segments to find the knife in whole image of resolution 640 x 480 pixels.

Time required to detect both human and knife, for knife detection after human detection using sliding window and neural network classification is 0.90 sec. Number of image segments scanned in this method were 240.

Time required for computation of knife detection using background subtraction segmentation is 0.58 sec. Number of image segments required to search were 138 in considered image and changes depending on background subtraction results. Figure 5.15 shows the reduction in false classifications when human is detected before knife detection. Green boxes represent true positives and red boxes represent false positives.
Figure 5.15: Comparison of outputs of three methods.

The procedure of finding human and then knife in the surroundings of human has resulted in improved hit rate (correctly classified as knife) compared to finding a knife in complete image taken by CCTV camera containing background. This method reduced the number of false detections as area to be scanned to search knife was reduced.

5.7 Discussion

In this thesis knife is searched in the window of size $100 \times 100$ pixels in an image of resolution $640 \times 480$ pixels, as the database available for training the neural network has images with $100 \times 100$ pixels resolution. Since HOG is selected as feature vector for describing the features of knife and as HOG parameters are not adjusted adaptively to the size of the image, as well as length of the input vector to the neural network must be constant, knife is searched in the real time image using sliding window of $100 \times 100$ pixels. We could use the smaller windows like $20 \times 20$ pixels, $50 \times 50$ pixels etc. as sliding window and resize the window to $100 \times 100$ pixels before finding the feature vector but the ratio of knife image to total $100 \times 100$ pixels image segment would be different from the trained images and the classifier might not be able to classify the images properly. The images in the database can be downsampled before finding the feature vector and training the neural network but as the knife is small, downsampling could affect the quality of feature vector being computed.

Since single class classifier is trained on $100 \times 100$ pixel knife images, design
of the multi class classifier also requires the images of gun with same resolution. The images downloaded from Google images were of higher resolution and were downsampled to $100 \times 100$ pixels before finding the feature vector and training neural network to classify multiple types of weapons.

For finding the ROI i.e human with knife in the whole image of $640 \times 480$ pixels resolution two methods were used, background subtraction and HOG method. Both methods have advantages and disadvantages. The number of false positives that might occur in the former method would be less compared to later as the amount of images need to be tested for the detection of knife would be limited after background subtraction compared to HOG method. On the other hand using HOG method, a human may not always be detected due to false negatives. In those cases knife detection algorithm would not be performed as human was not detected. In background subtraction the chances of detecting the humans would be higher as we are subtracting the background and testing the edges in the image. Some disadvantages of background subtraction would be that subtraction may fail if the persons clothes are very close to the background colour. Background subtraction method might not be usable in outdoors as when the cameras are used in outdoors the background changes constantly according to day light, movement of branches of trees, moving vehicles etc. In these situations HOG method performs well as HOG features would not get affected significantly by changes in background.

For HOG method of human detection, the algorithm was evaluated on INRIA database test set. Navneeth Dalal and Bill Triggs in [17] described the HOG features and classified the humans from INRIA database with an accuracy of $89\%$ using SVM classifier. In this thesis we designed the neural network that was able to classify the humans from the INRIA database test set with an accuracy of $94.67\%$.

In this thesis performance of the neural network to detect knife was calculated using 10 fold cross validation on the complete dataset. Authors in [2] have used four fold cross validation and used 10508 images with 2627 images in each fold. They have evaluated the results in terms of sensitivity and specificity. They were able to achieve sensitivity and specificity of $81.18\%$ and $94.9\%$. On the same dataset containing 12899 we have used 12830 images and obtained the sensitivity and specificity of $80.4\%$ and $85.7\%$. Though the specificity and sensitivity achieved by us is slightly less than authors in [2] we have performed our cross validation on more number of images compared to authors in [2]. They have tested their algorithm on 10508 images only and 2391 images were missing in their cross validation.

Accuracy of our proposed method was $84.6\%$ on test dataset provided in table 4.1. This might be due to various factors like HOG method might fail when knife is held in a position such that only one edge may appear in the image. HOG method even fails when the image of knife might immersed in the background. HOG is rotation variant hence we need to train the neural networks with as many
rotations as possible. Extra reasons of failure can be, when the knife is blurred for example the knife is moved very fast. Figure 5.16(a), 5.16(b), 5.16(c) shows the images where HOG might fail to classify the segment as positive.

![Different examples where HOG might fail](image)

(a) Blurred knife  (b) Immersed knife  (c) Knife with only one edge

Figure 5.16: Different examples where HOG might fail
6.1 Conclusions

Focus of the thesis was mainly on implementation of knife detection algorithm using different methods of region of interest segmentation. Implementation of knife detection algorithm on real time image of resolution $640 \times 480$ pixels was performed successfully. Detection of knife directly from the whole image was performed which was found to be not useful when knife alone was in the frame without the presence of a human. Also, the time taken to detect a knife in the whole image was long. Hence in other two methods human was detected prior to knife detection. Background subtraction (method 3) can be implemented in indoor environment where the background remains constant. In outdoor environment, HOG human segmentation (method 2) was more preferable. A method through which multiple types of weapons (gun and knife) can be classified using a single multi-class classifier was implemented. In any recent studies, no such method was implemented. Though the accuracy of this method was slightly lower when compared to single weapon classifier, it is generalized without a need to maintain different algorithms for different weapons.

Background subtraction method could detect knife quickly when compared to sliding window method. The time taken to detect a knife using background subtraction method is $0.58$ sec, time taken for knife detection after HOG method of human detection is $0.90$ sec and the time taken to detect knife using sliding window method is $0.93$ sec. This detection can be extended to many types of weapons.

Single class classifier is designed for classification of knife in this thesis and was able to achieve accuracy of $84.6\%$ and specificity and sensitivity of the classifier for dataset of $12830$ images were $85.7\%$ and $80.4\%$ respectively. Multiple classifier neural network designed for the classification of gun and knife was able to achieve an accuracy of $83.03\%$ on dataset described in table 3.2.

6.2 Scope for Future Work

This research work can be extended in different ways:
Chapter 6. Conclusions and Future Work

1. Both background subtraction method which is useful in indoor environment and HOG human segmentation method which is useful in outdoor environment can be integrated into a single method and can be generalized.

2. Sensitivity and specificity of the proposed algorithms need to be further improved for real time implementation.

3. The accuracy of multi class classifier can be further increased by decreasing false alarms.

4. For training and testing images of guns, we didn’t use real time low quality images since we couldn’t get access to such images. This can be further extended to real time images which would be useful in real time applications.

5. The algorithm was implemented on images and extended to recorded video by collecting sequence of frames to detect knife from video. It can be further extended to detection in live video. For implementation in live video, algorithm must work very fast.
Bibliography


