Identification of Human Motion Using Stationary Camera

BY

Rachana Modi

09MCE031

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May, 2011

Identification of Human Motion Using Stationary Camera

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Rachana Modi

09MCE031

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHMEDABAD-382481

May, 2011

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Computer Science and Engineering at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgement has been made in the text to all other material used.

Rachana Modi

Certificate

This is to certify that the Major Project entitled ***" Identification of Human Motion Using Stationary Camera"*** submitted by ***Ms. Rachana Modi (09MCE031)*** , towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Tejas Mehta
Assistant Professor,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. S.N.Pradhan
Professor and PG-Coordinator,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. D.J.Patel
Professor and Head,
Department of Computer Engineering,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. K.Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Abstract

Video surveillance is currently one of the most active research topics in the computer vision community. During motion, the surveillance system can detect moving objects and identify them as humans, animals, vehicles. This strong interest is driven by a wide spectrum of promising applications in surveillance system such as Military security, Public and commercial security, etc. Recognition of Human Motion (RHM) is proposed system which includes detection, feature extraction and recognition of people from image sequences involving humans. In proposed system frame differencing method is used for moving object detection and Neural Network is used for recognition of human motion. System uses half bottom part of the frames for generating more accurate result which is identify the moving object is human. Experimental results show that human motion can be correctly classified.

Acknowledgements

I would like to present this report on the dissertation work related to Identification of Human Motion Using Stationary Camera. I am very thankful to all those who helped me for the successful completion of the dissertation and for providing valuable guidance throughout the project work.

First of all I would like to thank Prof. Tejas Mehta, Assistant Professor, CSE Department, Institute of Technology, Nirma University, whose knowledge and expertise in Digital Image Processing has been of a great pillar of support for my work. He has shown keen interest in this dissertation work right from beginning and has offered a great motivating factor in outlining the flow of my work.

I would like to thanks Dr. S. N. Pradhan, Prof. In Charge M.Tech CSE, Institute of Technology, Nirma University, Ahmedabad whose keen interest and excellent knowledge base helped me to finalize the topic of the dissertation work. His constant support and interest in the subject equipped me with a great understanding of different aspects of the required architecture for the project work.

My kind and sincere thanks and gratitude to Dr. Ketan Kotecha, Director Institute of Technology, Nirma University, Ahmedabad for his continual kind words of encouragement and motivation throughout the Dissertation work. My sincere thanks and gratitude to Prof. D. J. Patel, Professor and Head Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his continual kind words of encouragement and motivation throughout the Dissertation work. The blessings of God, department faculty members and my family members makes the way for completion of major project. I am very much grateful to them.

- Rachana Modi

Contents

Declaration **iii**

Certificate **iv**

Abstract **v**

Acknowledgements **vi**

List of Tables **ix**

List of Figures **x**

1 Introduction **1**

1.1 General Overview 1

1.2 Motivation 3

1.3 Objective 4

1.4 Scope of work 4

1.5 Organization of thesis 4

2 Literature Survey **6**

2.1 Moving Object Detection using static background 6

2.1.1 Background Subtraction Method 6

2.1.2 Statistical methods 9

2.1.3 Temporal Differencing 12

2.2 Moving Object Detection using dynamic background 15

2.3 Moving Object Recognition 18

3 Proposed Method **21**

3.1 pre-level processing 22

3.1.1 Moving Object Detection with static background 22

3.1.2 Moving Object detection with dynamic background 26

3.2 Post-level processing 28

3.2.1 Employing Network for Pattern Recognition 28

4 Implementation **34**

4.1 Tools & Technique used 34

4.2 Neural Network 34

5 Results and Discussion **37**

5.1 Results for Moving Object Detection with static background 37

5.2 Results for Moving Object Detection with dynamic background 38

5.3 Results of Feature extraction 39

5.4 Results of neural network pattern recognition 40

5.4.1 Result of normal size of the frame 40

5.4.2 Result of half bottom part of the frame 44

6 Conclusion And Future work **48**

Web References **49**

References **52**

Index **52**

List of Tables

2.1 Comparative Analysis of object detection methods 14
2.2 Methods for moving object recognition 20

List of Figures

3.1 Overview of RHM System 22
3.2 System diagram of object detection 23
3.3 Conversion of average frame's blocks into binary values 30
3.4 Records of normal size frames 32
3.5 Walking patterns of the human 32
3.6 Records of half bottom part of the frames 33

5.1 Background Subtraction sample, (a) Current image (b) Gray scale (c) Detected region using Simple background subtraction (d) Detected region using Frame Differencing. 38
5.2 Result of moving background using frame differencing method 39
5.3 Result of moving background using codebook with frame differencing method 39
5.4 Extracted Bounding box for consecutive image frames 40
5.5 Average of elements in that block 40
5.6 Confusion Matrix of Training, Validation & Testing Phase 41
5.7 Performance plot of each Epochs 42
5.8 Training state of each Epochs 43
5.9 ROC curves for each of the four categories 44
5.10 Confusion Matrix of Training, Validation & Testing Phase 45
5.11 Performance plot of each Epochs 46
5.12 Training state of each Epochs 46
5.13 ROC curves for each of the four categories 47

Chapter 1

Introduction

Computer vision [1]has been an active area of research for more than three decades. Its goal is to develop processing and visual sensing algorithms and hardware that can see and understand the world around them. Current applications, such as medicine, document analysis, remote sensing, autonomous navigation, as well as object (such as vehicles, animals) and human tracking systems, reveal that image processing and pattern recognition have been tremendously successful in terms of delivering operational systems. They have been used to describe an image or video sequence in terms of meaningful objects that comprise it. The task to classify and identify objects in the frame is surprisingly difficult for a computer program.

1.1 General Overview

Aim of human motion detection and recognition for the surveillance system. Video surveillance systems have long been in use to monitor security sensitive areas. The history of video surveillance consists of three generations of systems (generation surveillance systems) which are called 1GSS, 2GSS and 3GSS [1].

The first generation surveillance systems (1GSS, 1960-1980) were based on analog signal sub systems for image acquisition, transmission and processing. They extended human eye in spatial sense by transmitting the outputs of several cameras monitoring a set of sites to the displays in a central control room are used to present continuous flow visual sequences to operator. They had major drawbacks like requiring high bandwidth, difficult archiving and retrieval of events due to large number of video tape requirements and difficult online event detection, which was dependent on human operators with limited attention.

The next generation surveillance systems (2GSS, 1980-2000) were hybrids in the sense, that they used both analog and digital sub systems to resolve some drawbacks of its predecessors. They made use of digital video processing methods that provide assistance to the human operators by filtering out spurious events. Most of the work during 2GSS was focused on real-time event detection.

Third generation surveillance systems (3GSS, 2000 onwards) provide end-to-end digital systems. Image acquisition and processing at the sensor level, communication through mobile and fixed heterogeneous broadband networks and image storage at the central servers benefit from low cost digital infrastructure.

Unlike previous generations, in 3GSS some part of the image processing is distributed towards the sensor level, by the use of intelligent cameras that are able to digitize and compress acquired analog image signals and perform image analysis algorithms like motion and face detection with the help of their attached digital computing components.

The ultimate goal of 3GSS is to allow video data to be used for online alarm generation to assist human operators and for offline inspection effectively. To achieve this goal, 3GSS will provide smart systems that are able to generate real-time alarms defined on complex events and handle distributed storage and content-based retrieval of video data.

The making of video surveillance systems ”smart” requires fast, reliable and robust algorithms for moving object detection and tracking.

Moving object detection is the basic step for further analysis of video. It handles segmentation of moving objects from stationary background objects. This not only creates a focus of attention for higher level processing but also decreases computation time. The next step in the video analysis is recognition, which can be simply identify the detected objects from frame to frame.

The output of the system can be used for providing the human operator with high level data to help him to make the decisions more accurately in a shorter time.

Below are some scenarios that require recognition of human motion systems.

Public and commercial security

- Monitoring of banks, department stores, airports, museums, stations, private properties and parking lots for crime prevention and detection.
- Patrolling of highways and railways for accident detection.

Surveillance: Automated Monitoring of Activity in Scenes

- Monitoring traffic and related events such as accidents and other unusual events.
- Detecting people, their activities, and related events such as over staying.

Military security:

- Patrolling national borders.
- Measuring flow of refugees.

The use of object detection and recognition algorithms are not limited to video surveillance only. Other application domains also use these algorithms. Some examples are video compression, human machine interface, video editing and multimedia databases.

1.2 Motivation

The motivation behind doing this research is the need to develop a system for human motion detection & identification. This system provides security at Public and commercial area like Monitoring of banks, department stores, airports, museums, stations, private properties & parking lots for crime prevention and detection, for patrolling borders & Measuring flow of refugees which provides military security etc.

1.3 Objective

The objective of this project is to develop a system for the purpose of detecting motion (specifically moving human) from a video sequence captured by stationary camera.

A suitable technique of image processing is to be implemented for the purpose of detecting motion that exists in the video sequence. Secondary technique that is to be implemented for recognizing the detected object.

1.4 Scope of work

This project deals with the problems of defining and developing the basic building blocks of an automated recognition of human motion system. Initial problem is the detection of object motions in the scene. Frame differencing Background subtraction algorithm is capable of detecting the moving object with less amount of noise. The scope of this study would be restricted to mentioned boundaries.

- The scene does not include night vision and any weather condition.
- System's attached camera should be stationary.

1.5 Organization of thesis

This thesis covers the different methods to detect and recognize moving object using stationary camera for understanding level to the research level.

Chapter 2, covers a literature survey for different technique of moving object detection and recognition from video sequence for RHM system.

Chapter 3, includes one proposed system for human motion detection & recognition which is known as RHM system.

Chapter 4, describes about tools and techniques which is used for implementation of the RHM system.

Chapter 5, shows the experimental results of the RHM system.

Chapter 6, covers the conclusion & future works respectively.

Chapter 2

Literature Survey

This chapter covers the related work and the latest studies in the literature on each of these building blocks. There have been a number of surveys about object detection & recognition using static cameras in the literature. The survey we present here covers only that work which is same context as our study. Brief information on some techniques are discussed, which are used for similar tasks that are not covered in our study.

2.1 Moving Object Detection using static background

Detecting the moving objects from a video sequence is a fundamental and critical task of moving object detection. It plays a major role in advanced security systems and video surveillance applications. There are different methods to detect the foreground object from the background [4].

2.1.1 Background Subtraction Method

Background subtraction [2] is a particularly popular method for motion segmentation, especially under those situations with a relatively static background. It attempts to detect moving regions in an image by differencing between current image and a reference background image in a pixel-by-pixel fashion.

Drawback: It is extremely sensitive to changes of dynamic scenes due to lighting and extraneous events.

There are different approaches to this basic scheme of background subtraction in terms of foreground region detection, background maintenance and post processing.

a. **Heikkila and Silven** [3] uses the simple version of this scheme where detect moving regions by subtracting the current image pixel-by-pixel from a reference background image. The pixels where the difference is above a threshold are classified as foreground. As a result, a binary image is formed where active pixels are labeled with "1" and non-active ones with "0". If a pixel at location (x, y) in the current image, it is marked as foreground if

$$|\mathrm{I}_t(\mathrm{x},\ \mathrm{y}) - \mathrm{B}_t(\mathrm{x},\ \mathrm{y})| > \mathrm{T}$$

Where T is a predefined threshold.

Drawback: It is necessary to update the background image frequently in order to guarantee reliable motion detection.

The basic idea in background adaptation is to integrate the new incoming information into the current background image using the following first order recursive filter:

$$\mathrm{B}(\mathrm{k} + 1) = (1 - \alpha)\mathrm{B}(\mathrm{k}) + \alpha\ \mathrm{I}(\mathrm{k})$$

Where α is a coefficient. The larger it is, the faster new changes in the scene are updated to the background frame. However, cannot be too large because it may cause artificial "tails" to be formed behind the moving objects.

Two background corrections are applied:

- If a pixel is marked as foreground for more than m of the last M frames, then the background is updated as $\mathrm{B}_t+1 = \mathrm{I}_t$. This correction is designed to compensate for sudden illumination changes and the appearance of static new objects.

- If a pixel changes state from foreground to background frequently, it is masked out from inclusion in the foreground. This is designed to compensate for illumination, such as swinging branches.

Advantage: This algorithm could handle some of the inconsistencies due to lighting changes, etc.

b. **Yang and Levine** [4] proposed an algorithm based on the observation that the median value was more robust than the mean value, They constructing the background by taking the median value of the pixel color over a series of images. The median value, as well as a threshold value determined using a histogram procedure based on the least median squares method, was used to create the difference image.

Advantage: This algorithm could handle some of the inconsistencies due to lighting changes, etc.

c. **Karman, Brandt and Cappellini** [5] observes and predicts RGB values for each background pixel, based on previous observations using Kalman filter. A Kalman filter is very widely used recursive technique. A Kalman based technique tracks linear dynamical systems under Gaussian noise. It is a simple technique under an assumption that the intensity values of a pixel can be modelled by a probability distribution (e.g. Gaussian distribution). Here the models are updated using simple adaptive filters (e.g. IIR filters) which can accommodate changes in lighting or objects that become part of background. Karmann and von Brandt use both intensity and temporal derivative. The internal state of the systems is described by its background intensity Bt and its temporal derivative is defined as B't. A brief description of scheme used in [5] is described below:

$$\begin{bmatrix} B_t \\ B'_t \end{bmatrix} = A. \begin{bmatrix} B_t - 1 \\ B'_t - 1 \end{bmatrix} + K_t. \left(I_t - H.A. \begin{bmatrix} B_t - 1 \\ B'_t - 1 \end{bmatrix} \right)$$

Background dynamics is described by the matrix A and H is the measurement matrix. The Kalman gain switches between a slow adaptation rate $\alpha 1$ and high adaptation rate $\alpha 2$.

$$K_t = \begin{bmatrix} \alpha 1 \\ \alpha 2 \end{bmatrix} if I_t is foreground, and \begin{bmatrix} \alpha 2 \\ \alpha 1 \end{bmatrix} \text{ Otherwise}$$

2.1.2 Statistical methods

These statistical methods are mainly inspired by the background subtraction methods in terms of keeping and dynamically updating statistics of the pixels that belong to the background image process. Foreground pixels are identified by comparing each pixel's statistics with that of the background model.

Advantage: Its reliability in scenes that contain noise, illumination changes and shadow.

a. **Haritaoglu** proposed W4 [6] system uses a statistical background model where each pixel is represented with its minimum (M) and maximum (N) intensity values and maximum intensity difference (D) between any consecutive frames observed during initial training period where the scene contains no moving objects. A pixel in the current image, it is classified as foreground if it satisfies:

$$|\mathrm{M(x, y)} - \mathrm{It(x, y)}| > \mathrm{D(x, y)} \text{ or } |\mathrm{N(x, y)} - \mathrm{It(x, y)}| > \mathrm{D(x, y)}$$

After thresholding, a single iteration of morphological erosion is applied to the detected foreground pixels to remove one-pixel thick noise. In order to grow the eroded regions to their original sizes, a sequence of erosion and dilation is performed on the foreground pixel map. Also, small-sized regions are eliminated after applying connected component labeling to find the regions. The statistics of the background pixels that belong to the non-moving regions of current image are updated with new image data.

b. **Wren** proposed another statistical method [7], a simplest form of background estimation is to calculate the average image of scene, subtract each incoming frame with the average and threshold the result. A basic single Gaussian model can adapt to slow changes in the scene by recursively updating the model using a simple adaptive filter. This simple adaptive approach has been used in "pfinder system" for tracking human body. The main advantage with such a method is that it ignores the order in which observations are made and uses the distribution of pixel intensities. The drawback with a unimodal approach is that they do not adapt to changing background as each pixel, and does not deal with relocation of background object.

c. **McFarlane and Schofield** [8] propose an adaptive median filter method to estimate the median of pixel intensities for each frame. This kind of Gaussian model can adapt to slow changing background as it takes an average image of the scene and subtracts the current frame with the average.

The only disadvantage with this approach is that it adapts slowly towards a large change in the background and it needs several frames to learn the new background.

d. **Oliver** [9] proposed an approach of Eigen backgrounds. It is also based on eigenvalue decomposition, but this time applied to the whole image instead of blocks. Such an extended spatial domain can extensively explore spatial correlation and avoid the tiling effect of block partitioning. The method can be summarized as follows:

Laming phase:

- A samples of n images is acquired, each image with p pixels; the average image, μb, is then computed and all images mean-subtracted;
- The covariance matrix is computed and the hest M eigenvectors stored in an eigenvector matrix, ϕ_{Mb}, of size M x P.

Classification phase:

- Every time a new image I, is available, it is projected onto the eigenspace as
  I' = ϕ_{Mb} (I- μb)
- I' is then back projected onto the image space as
  I" = ϕ_{Mb}^T I' + μb
  Since the eigenspace is a good model for the static parts of the scene, but not for the small moving objects, I" will not contain any such objects.
- Foreground points are eventually detected at locations where $|I - I"| >$ T.

e. **Seki** [10] try to go beyond the idea of chronological averages by exploiting spatial co occurrence of image variations. Their main statement is that neighboring blocks of pixels belonging to the background should experience similar variations over time. Although this assumption proves true for blocks belonging to a same background object (such as an area with tree leaves), it will evidently not hold for blocks at the border of distinct background objects (this is likely the cause of several false detections appearing at the borders of different background objects).

The method in [10] can be summarized as follows:

Instead of working at pixel resolution, it works on blocks of N x N pixels treated as an N^2-component vector. This trades off resolution with better speed and stability.

Learning phase:

- For each block, a certain number of time samples is acquired; the temporal average is first computed and the differences between the samples and the average are called the image variations;
- The N^2 x N^2 covariance matrix is computed with respect to the average and an eigenvector transformation is applied reducing the dimensions of the image variations from N^2 to K.

Classification phase for the current block

- A neighboring block, u, is considered, with its current input value; the corresponding current eigen image variation is computed, called Z_u;

- The L-nearest neighbors to Z_u in the eigenspace, $Z_{(u,i)}$, are found and Z_u expressed as their linear interpolation;
- The same interpolation coefficients are applied to the values of the current block, b, which have occurred at the same time of the $Z_{(u,i)}$, this provides an estimate Z'_b for its current eigen image variation Z_b;
- The rationale of the approach is that Z_b and Z_b^* should be close if b is a background block; to measnre closeness, a cumulative probability over the 8-neighbouring blocks is used.

2.1.3 Temporal Differencing

Temporal differencing method detects moving regions by use of the pixel-by-pixel difference of consecutive frames (two or three) in a video sequence. This method is highly adaptive to dynamic scene changes; however, it generally fails in detecting whole relevant pixels of some types of moving objects. This method fails to detect stopped objects in the scene.

a. **Lipton** [11] detected moving targets in real video streams using temporal differencing. This method takes consecutive video frames and determines the absolute difference. A threshold function is then used to determine change. If I_n is the intensity of nth frame, then the pixel wise difference function Δn is satisfy the following equation are marked as foreground.

$$\Delta\mathrm{n} = |\mathrm{I}_n - \mathrm{I}_{n-1}|$$

And a motion image M_n can be extracted by thresholds.

$$Mn(u,v) = \begin{cases} In(u,v) & \Delta n(u,v) \geq T \\ 0 & \Delta n(u,v) < T \end{cases}$$

b. **Collins** developed a hybrid method that combines three-frame differencing with an adaptive background subtraction model [12]. The key idea is to maintain an evolving statistical model of the background to provide a mechanism that adapts to slow changes in the environment. For each pixel value P_n in the n^{th} frame, a running average P_n and a form of standard deviation P_n are maintained by temporal filtering, implemented as:

$$\bar{P}_{n+1} = \alpha P_{n+1} + (1-\alpha)\bar{P}_n$$

$$\bar{\sigma}_{n+1} = \alpha |P_{n+1} - \bar{P}_{n+1}| + (1-\alpha)\bar{\sigma}_n$$

where $\alpha = \tau$ * f , f is the frame rate and τ is a time constant specifying how fast (responsive) the background model should be to intensity changes. The influence of old observations decays exponentially over time, and thus the background gradually adapts to reflect current environmental conditions. A pixel has a value which is more than 2σ from P_n, then it is considered a foreground pixel.

The hybrid algorithm successfully segments moving regions in video without the defects of temporal differencing and background subtraction.

Comparative Analysis of described object detection methods are shown in following table 2.1.

	Background Subtraction	Statistical Method	Temporal differencing
How it works?	It attempts to detect moving regions in an image by differencing between current image and a reference background image in a pixel-by-pixel fashion.	Foreground pixels are identified by comparing each pixel's statistics with that of the background model.	Temporal differencing method detects moving regions by use of the pixel-by-pixel difference of consecutive frames (two or three) in a video sequence.
Who one used it?	1. Heikkila and Silven*(Adaptive background subtraction)* 2. Yang and Levine *(Median Filtering)* 3. Karmann, Brandt and Cappellini *(Kalman Filter)*	1. Haritaoglu *(Minimum-Maximum Filter)* 2. Wren *(Single Gaussian)* 3. McFarlane and Schofield *(Approximate Median Filter)* 4.Oliver *(Eigenbackgrounds)* 5. Seki *(Co occurrence of image variations)*	1. Lipton *(temporal differencing)* 2. Collins*temporal differencing with adaptive background subtraction*
Comparative Analysis			
Noise	High	Less	Less
Time consumer	Low	Yes	yes
Frequently need to change background Image	yes	yes	No

Table 2.1: Comparative Analysis of object detection methods

2.2 Moving Object Detection using dynamic background

Dynamic background means background has little variation like leaves movement, shadow, atmosphere changing, running fountain etc. There are some existing methods to detect that variations and segments the foreground object, which describe below.

a. **Stauffer and Grimson** [13] described an adaptive background mixture model for real-time tracking. In their work, every pixel is separately modeled by a mixture of Gaussians. A Mixture of Gaussian (MoG) approach allows to model background variation using a number of Gaussian distributions. This approach is parametric and the model parameters can be adaptively updated without storing a large buffer of previous frames. In recursive techniques such as MoG, no inter-dependencies between image pixels are assumed. Although these methods have had success with slow-changing backgrounds, they do not perform well with dynamic motion of the background.

   MOG have some disadvantages: Backgrounds having fast variations are not easily modeled with just a few Gaussians accurately, and it may fail to provide sensitive detection (which is mentioned in [14]). In addition, depending on the learning rate to adapt to background changes, MOG faces trade-off problems. For a low learning rate, it produces a wide model that has difficulty in detecting a sudden change to the background. If the model adapts too quickly, slowly moving foreground pixels will be absorbed into the background model, resulting in a high false negative rate. This is the foreground aperture problem [15].

b. **Elgammal, Harwood and Davis**[14] use non-parametric method in their work. A non-parametric method models a pixel as a random variable in a feature space with an associated probability density function. These methods estimate the density function directly from the data without any assumption about the underlying distribution. This helps in avoiding choosing a model and estimating its distribution parameters. Kernel density estimators asymptotically converge to any density function. All non-parametric density estimation methods can be shown to be asymptotically kernel methods. Theoretically a non-parametric estimate can converge to any density shape with enough number of samples. The authors use the entire history of the incoming frames L1; L2; ::: Lm−1 to form a non-parametric estimate of the pixel density function f(Lt = u) :

$$f(Lm = u) = 1/L \sum_{i=t-L}^{t-1} K(u - Ii)$$

Here K(:) is a kernel estimator chosen to be a Gaussian. A new incoming pixel is termed as foreground if it is unlikely to come from the above distribution. In other words, f(Lm) is less than a threshold. A major issue that needs to be addressed when using kernel density estimation is the choice of kernel bandwidth. In theory as the number of samples reach infinity the choice of bandwidth is insignificant and the estimate will approach the actual density. But in practicality, only a finite number of samples are available and the computation must be performed in real time, hence, the choice of bandwidth is essential and above equation needs to be computed for every pixel.

Kernel method has drawback, it requires large memory usage to store value of each pixel to maintain the background model. So the processing speed is very slow.

c. **Kim, Chalidabhongse, Harwood and Davis** [5] develop codebook (CB) background subtraction algorithm for subtracting moving background without making parametric assumptions. The CB algorithm adopts a quantization/clustering technique, to construct a background model from long observation sequences. For each pixel, it builds a codebook consisting of one or more codewords. Samples at each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds. Not all pixels have the same number of codewords. The clusters represented by codewords do not necessarily correspond to single Gaussian or other parametric distributions.

Detection involves testing the difference of the current image from the background model with respect to color and brightness differences. If an incoming pixel meets two conditions, it is classified as background −(1) the color distortion to some codeword is less than the detection threshold, and (2) its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground.

The key features of the algorithm are:

- An adaptive and compact background model that can capture structural background motion over a long period of time under limited memory. This allows us to encode moving backgrounds or multiple changing backgrounds.
- The capability of coping with local and global illumination changes.
- Unconstrained training that allows moving foreground objects in the scene during the initial training period.
- Layered modeling and detection allowing us to have multiple layers of background representing different background layers.

2.3 Moving Object Recognition

Moving object recognition has been one of the active research areas in computer vision and pattern analysis. The aim of recognize moving objects through video monitoring system, it is classify these objects into predefined categories. Enhancement to current security systems and surveillance applications can be realized with this recognition function. For instance, intruder recognition function can be incorporated into a security system to classify intruders in order to reduce nuisance alarm and minimize human errors in manned surveillance system. Another example of its application is in traffic monitoring system, where it can be used to estimate traffic flow by making vehicle and pedestrian counts [2].

Extensive research efforts have been dedicated to moving object recognition, where many approaches have been presented to tackle this problem.

An overview of the related techniques is presented.

a. **View-based method** View-based method for recognizing 3D objects from 2D images [16] is exploited to serve this purpose. An aspect graph structure is implemented to generate aspects using a notion of similarity between views. A viewing sphere is endowed with a metric of dissimilarity for each pair of views. The viewing sphere is sampled at regular intervals, and the views are combined into aspects, each represented by a prototypical view. Unknown views of unknown objects are compared with the prototypical views hierarchically, and results are ordered by similarity. Two similarity metrics for shape were used, one based on curve matching and another based on matching shock graphs. This approach is significantly time consuming and memory intensive.

b. **Shape-based method** An approach adopted statistical motion detection and Fourier descriptors for shape-based moving object recognition [17]. A statistical, adaptive, illumination invariant motion detection algorithm is used to identify moving object candidates. Fourier descriptors [18] are computed as feature vectors to describe the shape of object. The classification module utilizes a four-layered feed-forward neural net. Objects are categorized as human, vehicle or background clutter.

c. **Motion-based method** Motion-based recognition method using Generalized Symmetry Operator to extract motion symmetry of moving objects for gait recognition [19]. First, Sobel operator is applied to the object silhouette to obtain edge map. A symmetry operator is then applied to the edge map to produce symmetry map. Fourier transform is applied to the gait signature (obtained by averaging all symmetry maps in an image sequence) and the k-Nearest Neighbor rule [20] is adopted for classification. It is relatively immune to noise and capable of handling occlusion. However, for practical purposes, the recognition rate on a larger database may diminish by selection of fewer Fourier components to improve recognition speed.

d. **Specific feature based method** A specific feature vector called Recurrent Motion Image (RMI) [21] is proposed to estimate repetitive motion behavior of moving objects. Different object has different motion behavior yielding different RMI. Thus, moving objects can be classified as single person, group of persons or vehicle based on their corresponding RMI. This approach starts with background subtraction and shadow removal, followed by region-based tracking to establish motion correspondence. Repetitive changes in the shape of object yield recurrent motion behavior which is used to generate the RMI. The areas of RMI demonstrating high motion recurrence will be used to determine the object's class. For example, the RMI of a walking human has high recurrence at the areas of hands and legs. This approach produces accurate classification results while remaining computationally and space efficient.

e. **Hybrid classification system** A hybrid classification system [22] can be used to recognize moving objects based on motion and appearance features simultaneously. At the first layer of the hybrid classifier, appearance data is processed by support vector machine (SVM) classifier [23], the resulting feature vectors are known as shape and appearance features. These features are combined with motion-based features, and used as input to the SVM classifier at the second layer.

Short description of Methods for moving object recognition shown in following table 2.2.

	View Based Method	Shape Based Method	Motion Based Method	Specific Feature Based Method	Hybrid Method
How it Works?	Make the view based on different aspect	Statistical Method + Fourier descriptor	Generalized Symmetry operator, which extract motion symmetry of moving object	Specific feature vector RMI is used, RMI estimate repetitive motion behavior of moving object	Based on motion and appearance feature
Classification algorithm	Decision Tree	Case based classification algorithm	K-mean data mining algorithm	RMI	Support Vector Machine

Table 2.2: Methods for moving object recognition

Chapter 3

Proposed Method

Proposed system is divided into 2 categories, namely Pre level processing (Detection) and Post level processing (Feature Extraction and Recognition) of object. Detection stage is able to distinguish moving foreground objects from static background objects in dynamic scenes. Feature Extraction stage extracts the features from the foreground objects and shows the bounding box around moving object. Recognition stage identifies the moving object from the scenes. In the following page we describe the flow of our approach system which is really helpful to reach the goals.

This RHM system is assumed to work real time as a part of a video-based surveillance system. The computational complexity and even the constant factors of the algorithms are important for real time performance. Furthermore, this system use is limited only to stationary cameras.

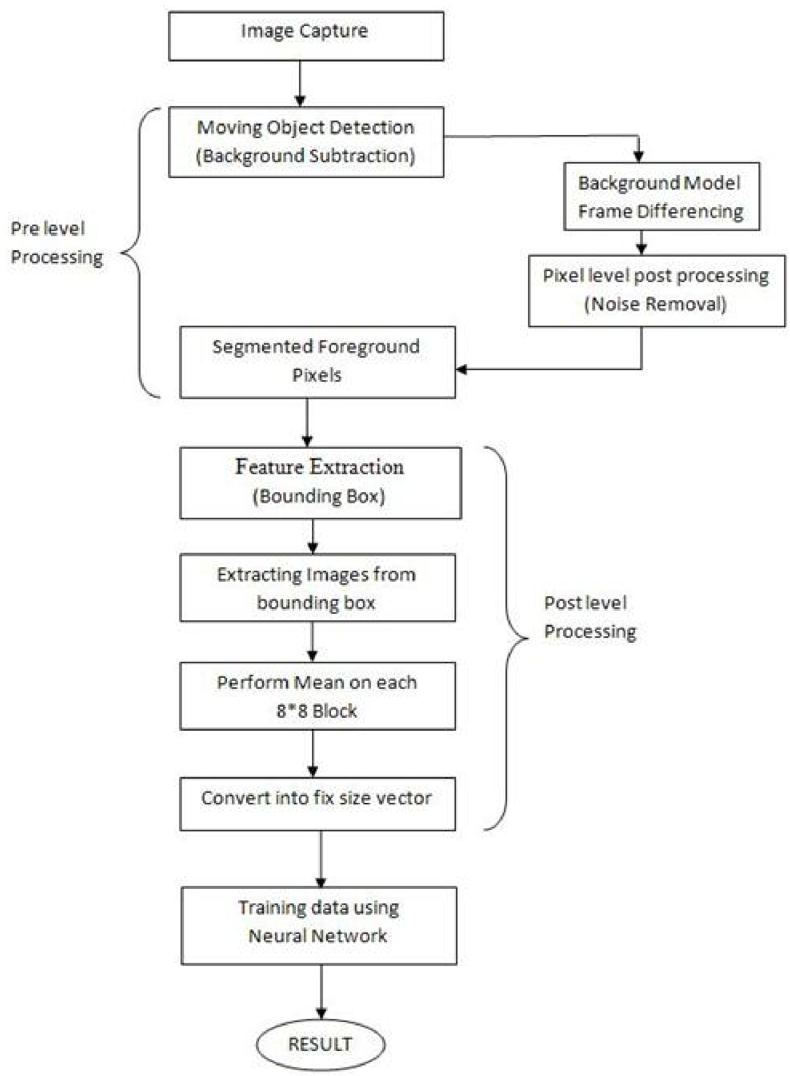


Figure 3.1: Overview of RHM System

3.1 pre-level processing

3.1.1 Moving Object Detection with static background

Distinguishing foreground objects from the stationary background is a difficult research problem. Object detection is the first step of the pre level processing which detects foreground objects in the sequence of frames. This creates a focus of attention for higher processing levels such as tracking, classification and behavior understanding and reduces computation time.

Normally, Simple background subtraction identifies moving objects by selecting the parts of the image which differ significantly from the background of the image. The proposed system's object detection algorithm follows a simple flow diagram shown in fig 3.2. In Background model, there is a statistical description of the current background scene and apply Frame differencing method, which generates output as a binary candidate foreground mask. Next step, Pixel level post processing is a noise removal technique which is applying on foreground mask. Finally, Foreground pixels are segmented from the foreground detection.

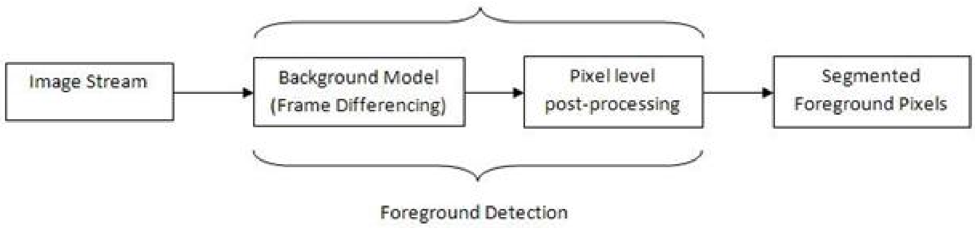


Figure 3.2: System diagram of object detection

a. **Image Stream**

The system retrieves images from the video (25 frames per second) file. The system will start to initialize the background using the first frame. The captured frames are converted to grayscale images from the RGB.

A grayscale image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other color image is that less information needs to be provided for each pixel. In fact a 'gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a image. The grayscale intensity is stored is an 8-bit integer giving 256 possible different shades of gray from black to white. Grayscale images are sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images. There is a drawback which is same intensity values are removed in background detection.

b. **Background Model**

There are some methods all ready exist to detect the foreground objects such as simple background subtraction, frame differencing etc. In Simple Background subtraction method identifies moving objects by selecting the parts of the image which differ significantly from a reference image, result of that shown in fig 5.1(c) while frame differencing [24] takes the difference between two consecutive frames and identifies moving objects. Comparing these two methods and shows that in frame differencing method has less noise generated, so we choose frame differencing method for subtracting background.

Frame Differencing Background Subtraction Model

For detecting the foreground region, system use Frame Differencing Background Subtraction Method, which really works in outdoor environment as well as indoor environment [24]. The Frame differencing identifies moving objects from the consecutive frames that differs significantly from the previous frame.

In this method there are two images as input. It produces output image in which previous image's pixel values are subtracted from the current image's pixel values. The output pixel values are given by:

The Algorithmic steps are given as below for Object Detection Module:

(1) Take Current frame I_{curr}.

(2) Retrieve the Previous frame I_{prev} from an array of image variables which are used for temporary storage of frames. The array is programmed to behave as a queue with only three elements at any given point of execution.

(3) Perform Frame Differencing Operation on the current frame I_{curr} and the previous frame I_{prev} where the resultant image is represented as output Image I_{out}.

$$I_{out} = I_{curr} - I_{prev}$$

(4) Perform the Binary Thresh-holding Operation on resulting image which is separating the pixels corresponding to the moving object from the background and set pixels as a '1' for moving object.

If value of pixels in input image is greater than value of threshold T, then the pixel value of binary image set as 1 (white) and remaining pixels set as 0 (black). Based on experiments, threshold value is set to 0.2. This approach gives fairly well results which shown in fig 5.1(d).

c. **Pixel level post processing**

The output of frame differencing algorithm contains some amount of noise that cannot be handled by the background model. There are various factors that cause the noise in foreground detection such as: Camera noise, Reflectance noise and background colored object noise. This noise affects the outputs of many calculation stages during the processing of a frame and the overall mask becomes inaccurate due to noise. In order to get improved results, noise removal is a crucial step. For this purpose, some techniques are used in the proposed system.

In pixel level post processing, perform an Iterative Mathematical Morphological Erosion Operation [25] on binary image to remove really small particles (noise). The result of this step is again a binary image. This step ensures that small insignificant movements in the background are ignored for better object detection. As a result, Foreground pixels are accurately segmented from the foreground detection.

After pre-level processing stage we know object is under motion and it is really helpful for further processing.

3.1.2 Moving Object detection with dynamic background

When background has little variation like leaves movement, The frame differencing method is not suitable because It does not subtract the little variation in a frame and also the border of moving object is not clear which is shown in fig 5.2. To overcome this problem, codebook based technique is used.

Various factors such as leaves movement, running fountain and Rainy atmosphere, which makes it difficult to segment foreground object. There are various methods have been described to handle moving background in literature survey. To handle moving background, Codebook mechanism is explored further. This concept is utilized to construct the codebook in proposed mechanism. The overall construction process is described as below.

A codebook covers Minimum & Maximum RGB values and Intensity values of each pixel to detect the moving object.

Codebook:

Main purpose is segmenting the foreground from the dynamic background. So for that system needs the database of background. Codebook is a set of codewords where as each codeword represents pixel information. The pixel consist minimum & maximum RGB values and Intensity values from the all frames. Detection involves testing the current image with respect to color and brightness combination. If an incoming pixel meets two conditions, it is classified as background – (1) The R, G, B value of current pixel comes in range of the corresponding pixel in a codebook (2) its brightness lies within the brightness range of that codeword. Otherwise, it is classified as foreground. The algorithm of codebook is described below:

Algorithm of codebook:

a. Codebook construction

(1) Take First pixel of all frames

(2) Find out R,G,B values of the pixel

(3) Compute the intensity using R,G,B values
I = $\sqrt{R^2 + G^2 + B^2}$

(4) Find out minimum value of R,G,B components and intensity value (I) from all frames

(5) Find out maximum value of R,G,B components and intensity value (I) from all frames

(6) Store in array

(7) Go to 1^{st} step and increase number of pixels.

This process is continue till last pixel of the frames and stored it in array which makes the codebook. Codebook has minimum and maximum values of R, G, B components and Intensity values of each pixel.

b. Codebook Testing

(1) Take new frame's pixel

(2) Codebook has range of the minimum & maximum value of the R, G, B values and intensity values of each pixel

(3) Trying to match the newer value of the pixel with the range of the corresponding pixel in a codebook

(4) If match is found then Set corresponding pixel = 0, which is known as background and match is not found then Set corresponding pixel = 1, which is known as foreground.

(5) Apply frame differencing algorithm on continuous frames.

Results of codebook with frame differencing is more suitable then the standard frame differencing method. Results are shown in fig 5.3.

3.2 Post-level processing

3.2.1 Employing Network for Pattern Recognition

Classification is one of the most active research and application areas of neural networks. Neural network is used as classifier for recognizing human. First step is to extract the features for training with neural network. Feature Extraction [26] plays a major role to detect the moving objects in sequence of frames for dimensionality reduction of data, so for that bounding box is used to visualize only moving object from the whole frame. To generate bounding box, Pixel values are found out from the first hit of the intensity values from top, bottom, left and right. By using this dimension values a rectangular bounding box is plotted within the limits of the values produced which explains a clear view on bounding box [27].

Algorithm for the bounding box is as followed

a. Read the image difference

b. for(present position = initial value: final value) of Y resolution

c. for(present position = initial value: final value) of X resolution

   (1) calculate the sharp change in intensity of image from top and bottom

   (2) store the values in an array

d. Height of the bounding box is = bottom value − top value

e. for(present position = initial value: final value) of X resolution

f. for(present position = initial value: final value) of Y resolution

   (1) calculate the sharp change in intensity of image from left and right

   (2) store the values in an array

g. Width of the bound box = right value − left value

h. Using the dimensions, draw boundary to the image Initial value: the starting position of the pixel in an image. Final value: the ending position of the pixel in an image.

$$\text{Height} = (\text{bottom value} - \text{top value}) \ / \ 2$$

$$\text{Width} = (\text{right value} - \text{left value}) \ / \ 2$$

i. Add the height value with the top value and store it in a variable like mid top

j. Add the width value to the left value and store it in a variable like mid left

k. Assign the max intensity to the pixel at pixel value at (mid top, mid left)

Using this process we get bounding box around the moving object which is shown in fig 5.4.

Next step is extracting the images from the Bounding Box. Each image is representing pixel values of the moving object. Bounding boxed image still contain too many pixels which in turn forms the larger input vector for neural network training. In order to reduce input vector length, boxed image is divided into 8*8 non -overlapping blocks, so for that taking the average value for each block results into smaller size vector and it is shown in fig 5.5.

After finding the average value of frame, we will get blocks of the average intensity values. Now we need to convert each block into binary pattern. For that, we have to predefine Threshold value (T = 10). Now we take index, if index value is more than 10 then it will give 1 otherwise it will 0. After that, we have patterns of blocks into binary values. It looks like following fig 3.3.

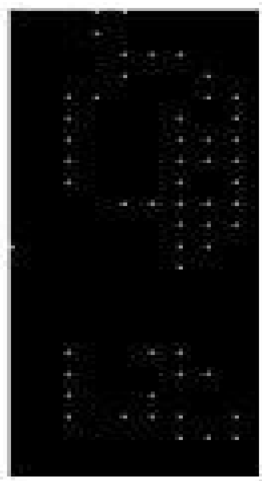


Figure 3.3: Conversion of average frame's blocks into binary values

After conversion each block results into fixed size vector. This vector forms a unique pattern for the various types of objects. This process is carried out for various other frames containing the different type of objects in motion. The inputs of image vector are fed simultaneously into the input layer of neural network.

For recognition, system has total 2205 patterns of various objects in input node. Each pattern has 345 attributes. There are 1543 patterns used for Training phase, 331 patterns are used for validation phase and rest of 331 patterns are used for testing phase. The experimental results show in following table 3.4 and fig 5.6 to 5.9.

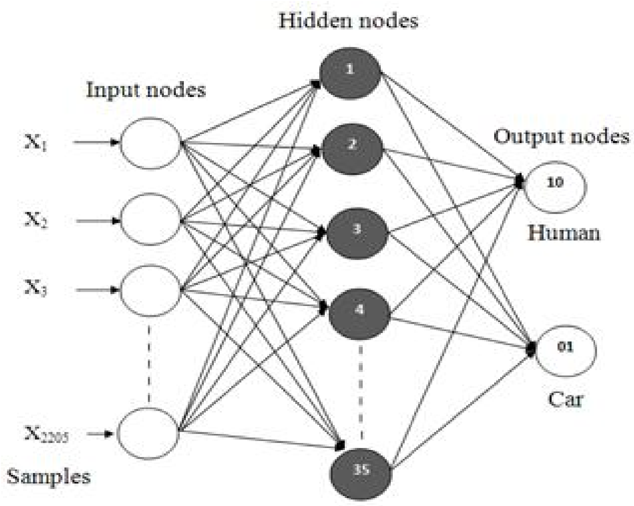


Table 1

Phase	Total Samples	Correctly Classified Samples		Miss Classified Samples	MSE	%E
		Human	Non Human			
Training Phase	1543	1161	151	231	$1.22013e^{-1}$	$14.97083e^{-0}$
Validation Phase	331	242	29	60	$1.35149e^{-1}$	$18.12688e^{-0}$
Testing Phase	331	271	22	38	$1.00862e^{-1}$	$11.48036e^{-0}$

Figure 3.4: Records of normal size frames

To generate more accurate results for human motion identification, half bottom part of the frames are used. The half part of the frame shows the leg pattern of human. When human is in walking condition, then patterns of legs are shown in fig 3.5.

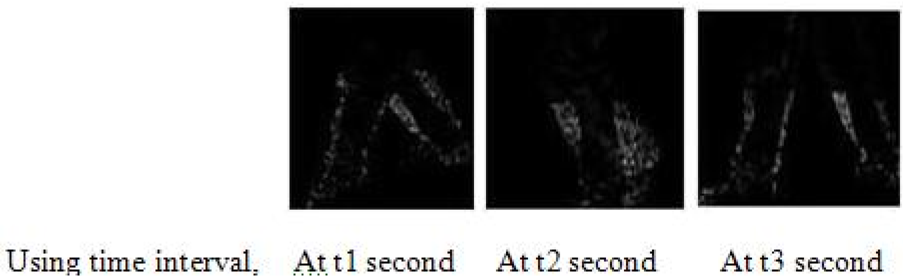


Figure 3.5: Walking patterns of the human

The Pattern of human walking is demonstrating repetitive motion. After some time interval it will be same. Other side we show the pattern of car is constantly same there is no change in pattern. There is an optional case, when car is turned.

Using time interval, system generates 3 different movements of leg. To combine all 3 movements, system gives the walking pattern of human. Now it is stored in smaller size of vector. This vector forms a unique pattern for the various types of objects. This process is carried out for various other frames containing the different type of objects in motion. The inputs of image vector are fed simultaneously into the input layer of neural network [28].

System has total 1258 samples with 545 attributes. Samples are divided into 3 phases: Training, Validation, and Testing phase which shows in Table 2. Thus, System identified 1094 samples correctly out of 1258 samples, so performance will be 87.0 percent shown in following table 3.6 and fig 5.10 to 5.13.

Table 2

Phase	Total Samples	Correctly Classified Sample		Miss-Classified Sample	MSE	%E
		Human	Not Human			
Training Phase	880	637	144	99	$8.89052e^{-2}$	$11.25000e^{-0}$
Validation Phase	126	89	18	19	$1.21109e^{-1}$	$15.07936e^{-0}$
Testing Phase	252	177	29	46	$1.39620e^{-1}$	$18.25396e^{-0}$

Figure 3.6: Records of half bottom part of the frames

Chapter 4

Implementation

4.1 Tools & Technique used

The videos used for this study were taken from the side view of the people and vehicles. Color videos were taken with a codec and sony digital cam with on uncompressed AVI formats. All video were taken at a 320 x 240 pixel resolution and 25 frames/second. For all experiments in this work, the algorithms were implemented in MATLAB 7.8 (R2009a). MATLAB provides a high-level programming environment specially tuned for matrix operations, which is a definite advantage in image processing tasks. MATLAB's Image Processing Toolbox v.6.0 provided a set of basic image processing functions, which were very helpful in the quick development of experimental modules.

MATLAB [3] provides inbuilt "Neural Network Pattern Recognition Tool" which is useful for recognition of human [28].

The test computer used for all experiments consisted of a personal computer with an Intel Core(TM) 2 Duo CPU at 2.00 GHz and 4 GB of RAM.

4.2 Neural Network

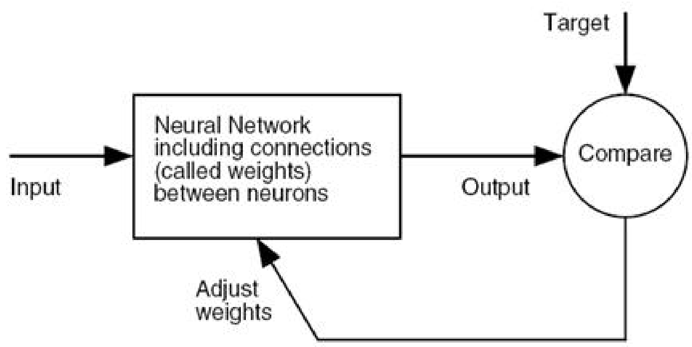
Neural network [7] works as the human brain. A neural network is an interconnected network of many artificial neurons. These artificial neurons are objects used to simulate the neurons in the human brain. Neural networks are used widely in machine learning applications. They are usually used for classification problems [29].

The recent vast research activities in neural classification have established that neural networks are a promising alternative to various classification methods. It is very important to understand how a neural network works. Neural network is a set of connected input/output units, where each connection has a weight associated with it. Neural network learns by adjusting the weights so as to be able to correctly classify the training data and hence, after testing phase, to classify unknown data.

Neural network pattern recognition tool

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The next figure illustrates such a situation. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically, many such input/target pairs are needed to train a network.



Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

Neural networks can also be trained to solve problems that are difficult for conventional computers or human beings. The toolbox emphasizes the use of neural network paradigms that build up to-or are themselves used in-engineering, financial, and other practical applications. The next sections explain how to use three graphical tools for training neural networks to solve problems in function fitting, pattern recognition, and clustering.

Chapter 5

Results and Discussion

5.1 Results for Moving Object Detection with static background

In this section, the results for moving object detection is presented and discussed. The sequence is obtained from a video with on no compression AVI formats. All video were taken at a 320 x 240 pixel resolution and 25 frames/second with a resolution of 320x240.

Sample foreground region detection is shown in 5.1. The first image shows the current frame, contains one foreground objects. The second image shows the gray scale of the original image. The third image and fourth image shows the detected foreground pixel map using normal background subtraction and frame differencing respectively. This Observation suggests that frame differencing gets better result than normal background subtraction.

Figure 5.1: Background Subtraction sample, (a) Current image (b) Gray scale (c) Detected region using Simple background subtraction (d) Detected region using Frame Differencing.

5.2 Results for Moving Object Detection with dynamic background

Background has little variation like leaves movement. The frame differencing method is applying but it is not suitable because It does not subtract the little variation in a frame and also the border of moving object is not clear which is shown in below fig 5.2.

Figure 5.2: Result of moving background using frame differencing method

Figure 5.3: Result of moving background using codebook with frame differencing method

5.3 Results of Feature extraction

Bounding box is used to visualize only moving object from the whole frame and following fig shows moving object of random selected frames.

Figure 5.4: Extracted Bounding box for consecutive image frames

Following figure shows the average value of boxed image is divided into 8*8 non - overlapping blocks and results into smaller size vector.

Figure 5.5: Average of elements in that block

5.4 Results of neural network pattern recognition

5.4.1 Result of normal size of the frame

a. **Confusion matrix**

Following figure shows generated results from neural network pattern recognition tool. It plots Confusion matrix for Training, Validation and Testing phase. Confusion Matrix shows corrected classified results and misclassified results of human and non human. The diagonal cells in each table show the number of cases that were correctly classified, and the off-diagonal cells show the misclassified cases. The blue cell in the bottom right shows the total percent of correctly classified cases (in green) and the total percent of misclassified cases (in red). The results for all three data sets (training, validation, and testing) show very good recognition. So overall results of correctly classifications are 85.1 %.

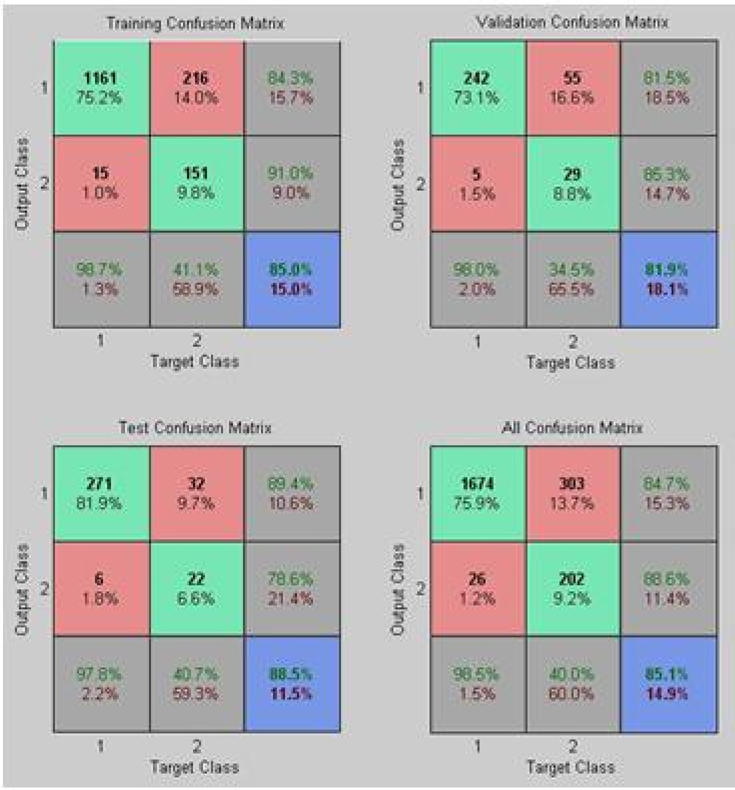


Figure 5.6: Confusion Matrix of Training, Validation & Testing Phase

b. **Performance Plot** Below figure shows the performance of the Training, Validation and Testing phases in different colors Blue, Green and Red respectively. The performance is measured by using Mean Squares Error (MSE) and number of Epochs (cycle).

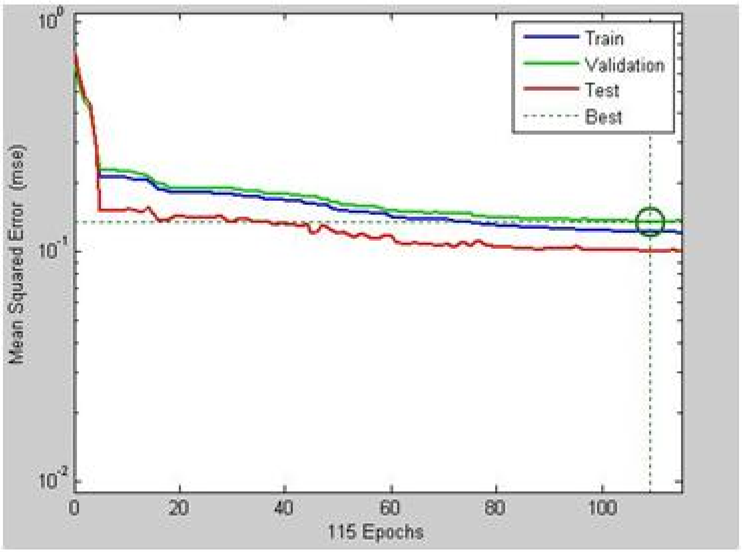


Figure 5.7: Performance plot of each Epochs

c. **Training state** It plots the training state from the training record, which returned by the trained network. It shows plot of gradient value Vs each epochs of the network.

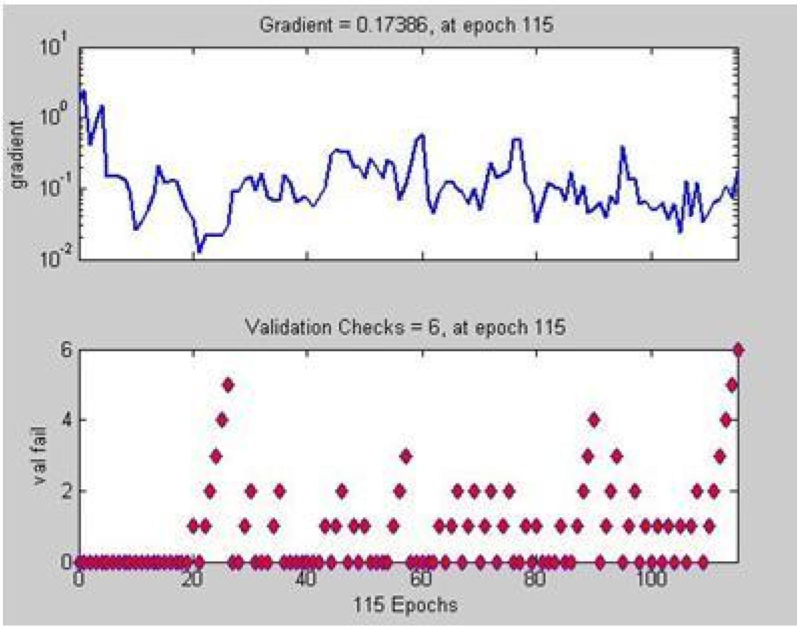


Figure 5.8: Training state of each Epochs

d. **Receiver operating characteristic** The colored lines in each axis represent the ROC curves for each of the four categories of this simple test problem. The ROC curve is a plot of the true positive rate (sensitivity) versus the false positive rate (1 - specificity) as the threshold is varied. A perfect test would show points in the upper-left corner, with 100

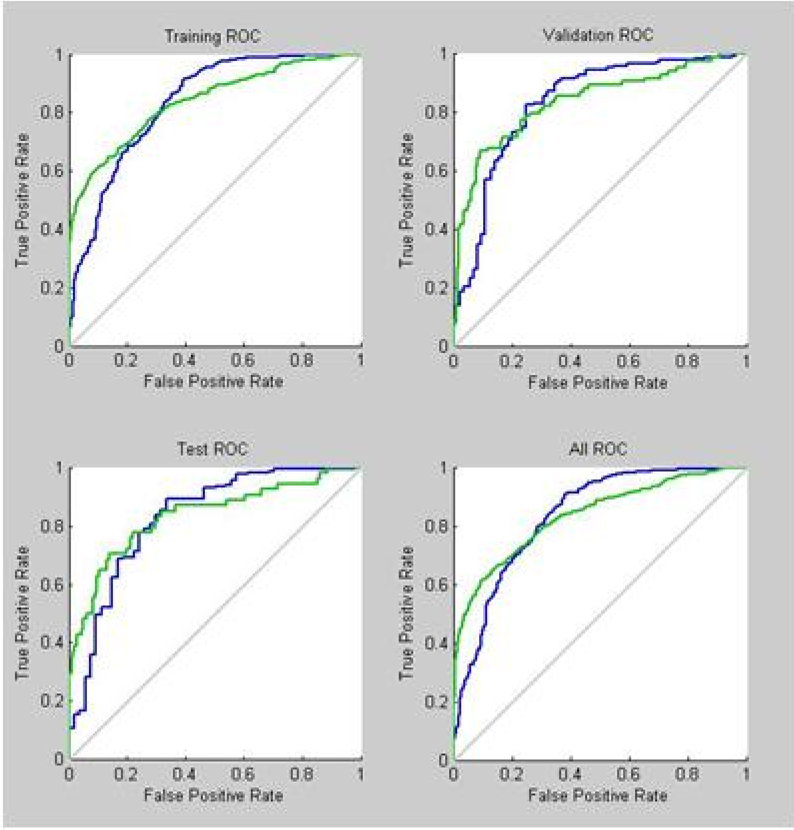


Figure 5.9: ROC curves for each of the four categories

5.4.2 Result of half bottom part of the frame

a. **Confusion matrix** Following figure shows generated results for half bottom part of the frames. Those inputs are giving to the neural network pattern recognition tool. It plots Confusion matrix for Training, Validation and Testing phase. Confusion Matrix shows corrected classified results and misclassified results of human and non human. So overall results of correctly classifications are 87.0%.

Figure 5.10: Confusion Matrix of Training, Validation & Testing Phase

b. **Performance Plot**

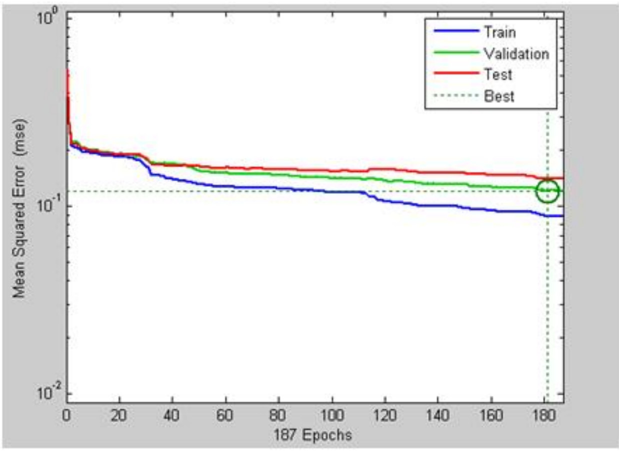


Figure 5.11: Performance plot of each Epochs

c. **Training state**

It plots the training state from the training record, which returned by the trained network. It shows plot of gradient value Vs each epochs of the network.

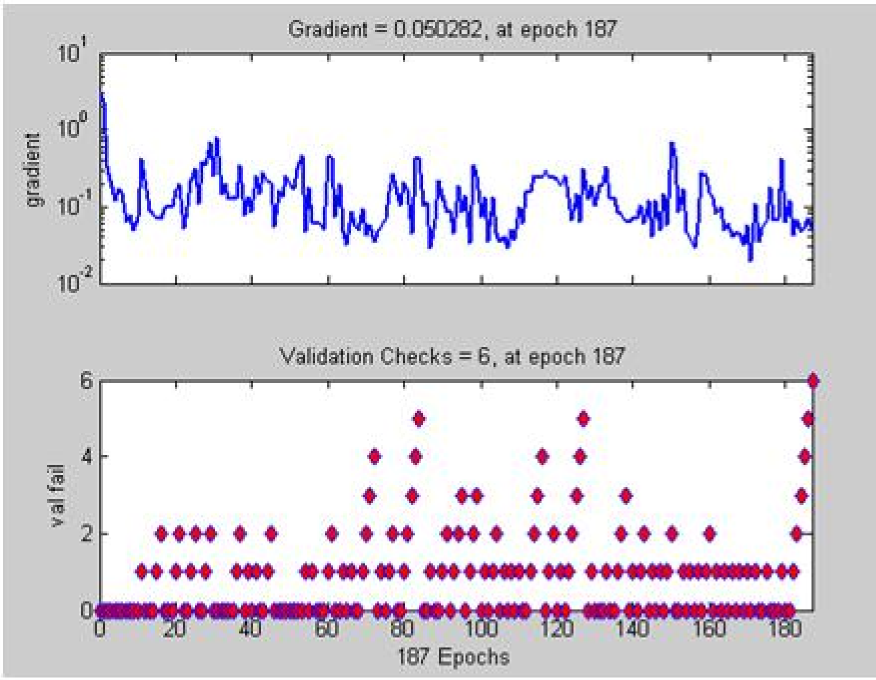


Figure 5.12: Training state of each Epochs

d. **Receiver operating characteristic**

The colored lines in each axis represent the ROC curves for each of the four categories of this simple test problem. The ROC curve is a plot of the true positive rate (sensitivity) versus the false positive rate (1 - specificity) as the threshold is varied. A perfect test would show points in the upper-left corner, with 100

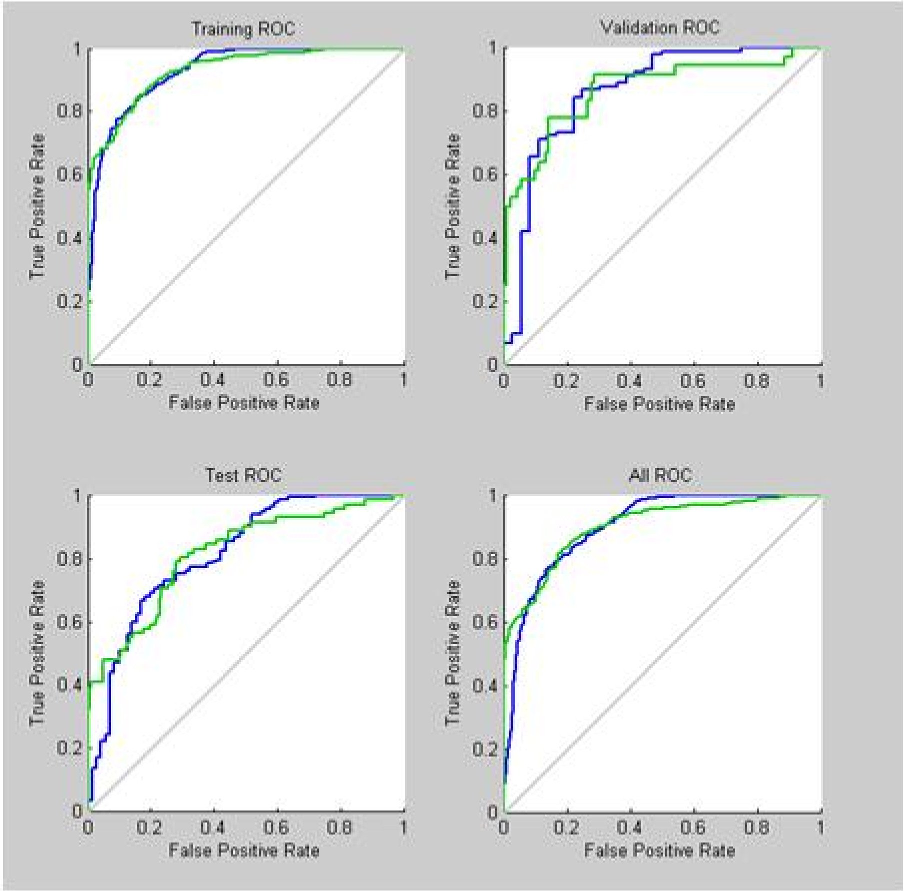


Figure 5.13: ROC curves for each of the four categories

Chapter 6

Conclusion And Future work

The proposed technique has been developed for video surveillance system using stationary camera. Pre processed results can be classified using classifier such as neural network and 85.1% accuracy is achieved. Analysis shows that considering bottom half part of frame results into 87.0% accuracy. Furthermore experimental result also shows that given technique try to deal with variations in the background.

Future work

Proposed work is suitable for static background as well as little variation in background. One of the first recommendations for future work is the incorporate with the constant moving background such as different weather condition, heavy traffic condition,etc. and Some special conditions such as shadow, reflectance, overlapping condition, object occlusion are still to be not handled. This handling of these condition would complete the work.

Web References

[1] http://en.wikipedia.org/wiki/Computer_vision.html

[2] http://www.research.ibm.com/peoplevision/bgs.html

[3] http://knol.google.com/k/matlab-image-processing#

[4] http://www.medianet.kent.edu/surveys/IAD01F-objdetection/index.html

[5] http://portal.acm.org/citation.cfm$?$id=1752979

[6] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.155.3335

[7] http://www.hoozi.com/post/ixsx7/introduction-to-neural-networks

References

[1] Pravesh kumar Goel (07MCE016), "Moving Object Detection and Tracking for Video Surveillance System", Nirma University Department Of Computer Science & Engineering, Ahmedabad, May 2009.

[2] L. Wang, W. Hu, T. Tan, "Recent developments in human motion analysis", Pattern Recognition, 36(3):585-601, March 2003.

[3] J. Heikkila, O. Silven, "A real-time system for monitoring of cyclists and pedestrians", In Proc. of Second IEEE Workshop on Visual Surveillance, pp. 74-81, Fort Collins, Colorado, June 1999.

[4] Y.H. Yang, M.D. Levine, "The background primal sketch: an approach for tracking moving objects", Machine Vision and applications, pp.17-34, May 1992.

[5] K.P. Karmann, A. Brandt,Cappellini V., "Moving object recognition using an adaptive background memory", Elsevier science publishers B.V., Amsterdam, The Netherlands, pp. 289-307, 1990.

[6] I. Haritaoglu, D. Harwood, L. Davis, "W4: A real time system for detecting and tracking people", In Computer Vision and Pattern Recognition, pp. 962-967, 1998.

[7] C. R. Wren, A. Azarbayejani, T. J. Darrell, A. P. Pentland, "Pfinder: Realtime tracking of the human body", IEEE Pattern Recognition and Machine Intelligence, 19(7):780-785, July 1997.

[8] J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications, 8:187-193, 1995.

[9] N.M. Oliver, B. Rosario,A.P. Pentland, "A Bayesian computer vision system for modeling human interactions", IEEE Trans. on Pattern Analysis. and Machine Intell., vol. 22, no. 8, pp. 831-843,2000.

[10] M. Seki, T. Wada, H. Fujiwara, d K. Sumi, "Background subtraction based on co occurrence of image variations", Proc. CVPR 2003, Vol. 2, pp. 65-72, 2003.

[11] A. J. Lipton, H. Fujiyoshi, R. Patil, "Moving target classification and tracking from real-time video", In Proc. of Workshop Applications of Computer Vision, pp. 129-136, 1998.

[12] Robert T. Collins , "A system for video surveillance and monitoring", Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May 2000.

[13] C. Stauffer, W. Grimson, "Adaptive background mixture models for realtime tracking", In Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 246-252, 1999.

[14] Elgammal A, Harwood D, Davis LS., "Non-parametric model for background subtraction", European Conference on Computer Vision 2000;2:751-67.

[15] Toyama K, Krumm J, Brumitt B, Meyers B., "Wallflower: principles and practice of background maintenance", International Conference on Computer Vision 1999; 255-61.

[16] C.M. Cyr, B.B. Kimia, "A Similarity-Based Aspect-Graph Approach to 3D Object Recognition", Intel J. Computer Vision, vol. 57, no. 1, pp. 5-22, 2004.

[17] D. Toth, T. Aach, "Detection and Recognition of Moving Objects Using Statistical Motion Detection and Fourier Descriptors", In Proc. 12th Intel Conf. Image Analysis and Processing (ICIAP 03), pp. 430-435, 2003.

[18] J.C. Russ, "The Image Processing Handbook", 5th ed., CRC Press, Boca Raton,p. 589, 2006.

[19] J.B. Hayfron-Acquah, M.S. Nixon, J.N. Carter, "Recognizing Human and Animal Movement by Symmetry", In Proc. IEEE Int'l Conf. Image Processing (ICIP 01), pp. 290-293, 2001.

[20] J.C. Russ, "The Image Processing Handbook", 5th ed., CRC Press, Boca Raton, p. 619, 2006.

[21] O. Javed, M. Shah, "Tracking and Object Classification for Automated Surveillance", In Proc.7th European Conf. Computer Vision-Part IV (ECCV 02),pp. 343-357, 2002.

[22] Y. Bogomolov, G. Dror, S. Lapchev, E. Rivlin, M. Rudzsky, "Classification of Moving Targets Based on Motion and Appearance", In Proc. British Machine Vision Conf. (BMVC 03),pp. 429-438, 2003.

[23] D.A. Forsyth, J. Ponce, "Computer Vision: A Modern Approach", Prentice Hall, Upper Saddle River, NJ, p. 615, 2002.

[24] Karan Gupta, Anjali V. Kulkarni, "Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching", Indian Institute of Technology, Kanpur, India.

[25] Edward R. Dougherty, "Mathematical Morphology in Image Processing", CRC Press, 1993.

[26] Feature extraction, Wikipedia (http://en.wikipedia.org/wiki/Feature_extraction)

[27] Bernhard Rinner, Wolfgang Schriebl, "Object Tracking and Velocity Determination using TMS320C6416T DSK", Internship Project Report, Klagenfurt University, April 2008.

[28] Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth, "Neural Network Toolbox 7 User guide", March 2009.

[29] Niyazpk, "Introduction to Neural networks", Hoozi Resources, March 2010.

Index

Abstract, v

Acknowledgements, vi

Certificate, iv