# Tracking and Abnormal Behavior Detection in Video Surveillance using Optical Flow and Neural Networks

**By**

**Nida Rasheed**

**2010-NUST-MS-PHD- ComE -12**
**MS-65 (ComE)**

**Submitted to the Department of Computer Engineering in fulfillment of the requirements of the degree of**

**MASTER OF SCIENCE**

**in**

**COMPUTER ENGINEERING**

**Thesis Supervisor**

**Dr. Shoab A. Khan**

**College of Electrical and Mechanical Engineering**

**National University of Sciences and Technology**

**2013**

**In the name of ALLAH the most Beneficent and Merciful**

# DECLARARTION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Brig. Dr. Shoab A. Khan. All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

**Nida Rasheed**

# DEDICATION

To our loving parents and teachers who made us learn how to lead this life, in stress and strain

# ACKNOWLEDGEMENTS

By the grace of Allah Almighty and continuous struggle of more than a year, I have been able to complete this project and present the final year project report. I am heartily thankful to our supervisor, Brig. Dr. Shoab A. Khan whose encouragement, guidance and support from initial to the final level enabled us to develop the understanding of the project.

His guidance helped me in all the time of research and writing of this report. I could not have imagined having a better supervisor for my final thesis. It goes without saying that if my worthy supervisor, Brig. Dr. Shoab A. Khan had not guide me with his scholarly guidance and instructions I would not have been able to complete this very important part of my degree.

I am also very thankful to my family who always supported me and prayed for me at every stage, without their love and concern I would have not been able to stand firm in my hard times.

For the completion of this project I always adopted a positive approach and that is the only reason we so far succeeded in achieving the goals.

Lastly, I offer my regards and blessings to Adnan Khalid, Lecturer Sajid Gul, Ali Jabir, Waqas Ahmad, Major Salman Rasheed and Dr. Imran Usman for their never ending support.

Nida Rasheed

# ABSTRACT

A target tracking algorithm that should also detect abnormal behavior detection is aimed to correctly identify the targets as being in a normal scenario or some chaotic movement. This dissertation is aimed for applications like highway traffic monitoring, railway stations, entry into restricted area etc. A model is developed here that presents the results in the form of normal and chaotic classes. The algorithm first extracts the information from the optical flow model using Lucas-Kanade approach. Optical flow model provides with the information of horizontal and vertical displacements of the objects of interest and the directions associated with each pixel. The features extracted from the model are then fed into neural network that is used for training as well as classification of the data. The uniqueness of this algorithm is that it uses foreground detection with Gaussian mixture model before passing the video frames to optical flow model. In this way the noise is being eliminated at the very initial stage and only the objects in motion are correctly identified. The study is being conducted on the real time videos taken from camera directly and some synthesized videos as well. The accuracy of method has been calculated by confusion matrix and mean square error of the neural network. The overall accuracy of the system is calculated as 97.5% and the percentage wrong classifications are 2.5%. The mean square error for the implementation was calculated as 3.5e-02.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| AI | Artificial Intelligence |
| BM | Background Model |
| BN | Bayesian Network |
| DNA | Deoxyribonucleic acid |
| FGMM | Foreground with Gaussian Mixture Model |
| FN | False Negative |
| FP | False Positive |
| FPE | Foreground Pixel Extraction |
| FPS | Frames per Second |
| GMM | Gaussian Mixture Models |
| HMM | Hidden Markov Model |
| ML | Machine Learning |
| MLE | Maximum Likelihood Estimate |
| MoG | Mixture of Gaussians |
| MSE | Mean Squared Error |
| MV | Machine Vision |
| NN | Neural Network |
| PDF | Probability Density Function |
| PF | Particle Filter |
| ROC | Receiver Operating Characteristics |
| TN | True Negative |
| TP | True Positive |
| CCTV | Closed-Circuit Television |

# Chapter 1: Introduction

## 1.1 Motivation and Objectives

Human needs and living standards have changed unbelievably with the evolution of world. Everything is becoming easy to achieve and at the same time, the competition to provide the easiest resources has increased the level of difficulty for the product manufacturers. Life has become too busy now a day. People need everything in their hands and they don't have the time to go through applications that take lots of time. With the advent of technology, applications like video surveillance's demand have been increased. This application took almost 60 years to become a common man's need or in other words, become accessible for general public. In past few decades, an application like this was associated with military areas only. Only some of the big and influential industries had the access to something like this as at that time, the cost of the cameras and equipment used for surveillance was very high. A sudden surge of advancement in this technology became prominent in the 1990's.

With the increasing requirements to reduce crime rates and to increase the security, many places like banks, stores etc became equipped with surveillance cameras and related machinery. Theft and other crime rates reduced noticeably in the areas with monitoring. In today's modern world, even the concept of monitoring has been changed. The cameras and machines are made intelligent enough to detect any abnormal behavior automatically and generate alarm or perform the predefined actions. These cameras can be embedded anywhere and do not need specialized equipment to support them. Another step ahead is that you do not have to be at the place in order to monitor a certain area. Video streams can be transmitted through internet or recorded for further use. Security in itself is a very big responsibility. Especially when it comes to surveillance of crowded places like airports and railway stations, data is huge; cost of processing that data is very high and timing is the most crucial aspect. Bandwidth is also an important factor as when the data is needed to be transmitted, it is needed that only the important information may be transferred and a minimum of bandwidth should be used for transmission of data.

At places like these, the behavior of crowd is studied. System is designed to investigate the abnormal behavior automatically. Example of this abnormal behavior is that there is a sudden chaos at the railway station. The system should be intelligent enough to decide that it's a normal behavior or something wrong has happened. Motion tracking is the eye candy for the researchers and application developers for many years. In busy environments like mentioned above, the system should be efficient, fast, reliable and of least cost. Different methods have been applied in this regard to achieve effective motion tracking and automatic detection of abnormal behavior. Thresholding, Template Matching, Histogram Based Tracking, Contour Based Tracking, Optical Flow techniques are the prominent ones in this field.

The research problem can be divided into two main parts. One is **Video Analysis** in which the features are extracted from the video for further processing. The second part is **Machine Learning** in which the extracted features are passed to the neural network tool. A neural network is created and trained with these extracted features.

These two parts of thesis are further divided into Motion Detection, Motion Recognition and Classification of movement as chaotic or non-chaotic.

## 1.2 Motion Detection

The first and foremost task in abnormal behavior detection in a video is to identify motion. Motion is a sign of change or sign of displacement in any scene. Different methods are further applied on the detected output to extract the information from the motion part only. The scene is segmented into motion and non motion parts. Motion detection is achieved by motion estimation and motion segmentation. Both are described below briefly.

### 1.2.1 Motion Estimation

Motion estimation is basically the estimate of displacement of moving objects from one frame to another. When a pixel moves from one position to another position, there are some features associated with that. These features are speed, direction, angle, magnitude, start point and end point. Same is the case for an object. The group of pixels create object that moves from one frame to another and all the above features are associated with the object. These features define the term motion as the change in them is an indicator that the object is being moved from one position to another. The motion is usually computed by comparing one frame with its subsequent frame. The crudest form of this is simple subtraction of frames where the difference indicates change in scene. The advanced and refined techniques include optical flow method, motion vector estimation and image subtraction using Gaussian filter.

### 1.2.2 Motion Segmentation

After estimating the motion, it is necessary to segment the video in the form of stationary or moving parts. The moving and stationary pixels are separated using different techniques so that further processing may be done on the video. There are different methods available to classify data as moving or non-moving and depending upon the application, the methods change. Clustering, Edge Detection, Histogram-Based, Level Set, Thresholding are among various techniques of motion segmentation. These techniques can either be adaptive or non adaptive.

## 1.3 Motion Tracking

Once the object is being identified and labeled as that this object is the one that is being displaced or moved, the next step that comes is to track the object. The object that is in motion can be tracked by using the feature set matching, blob tracking, kalman filter, template matching, particle filters. For abnormal behavior detection, prediction is also needed which comes in the next section.

## 1.4 Motion Recognition and Classification

After the object is identified as being in motion, the next step that comes is to recognize the type of motion that to which type of motion, the behavior is associated. These types may belong to circular motion, straight line motion, haphazard motion etc. The features extracted (speed, direction, magnitude etc) are then fed to a classifier for training and learning or testing. If this is the first time for the application to run for a certain area, the data is first extracted and trained for later classification. If the system is already in process, the features of the test video are matched with the existing data base to check the classification. Rule Based Classifiers, Bayesian Networks, Neural Networks, Hidden Markov Models, k-nearest neighbor etc are different classifiers that are used widely for classification.

## 1.5 Applications of Research

As discussed in start, this research is useful for areas where security is a very sensitive issues and a system is required that should be

- Intelligent to automatically detect the anomaly
- Least computations required

- Least use of resources required to store and process the data
- Error free and reliable
- Cost Effective
- Fast according to real time application

The research can be applied to many practical areas for security like:

- Airports
- Railway stations
- Building's entry and exit (Shopping Malls, libraries)
- Sports arenas
- Military areas

The application can also be used to monitor and track theft, accidents etc for:

- Vehicles on a highway to see if the traffic is in correct order
- Cars or bikes in a race to predict and prevent accidents
- Parking lots or banks to prevent theft or any other crime

The application is used for machine learning in:

- Gesture Recognition
- Gaze Tracking
- Complex Action Recognition
- Games Development
- Robots Development to Mimic Human Behavior
- Mood Recognition (Facial Expressions Recognition and Classification)

## 1.6 Scope and Limitations

The aim of this research is to create a system that can automatically recognize the behavior of the targeted area. It is achieved by using Foreground extraction technique, Gaussian smoothing, Median Filtering, Optical Flow using Lucas-Kanade Method, extracting the vertical and horizontal components separately, extracting rest of the features, classifying using neural network and simulating for different test cases. The prototype that is developed for the system is tested on sample videos.

The model takes the video as input that is smaller in resolution. For test purpose, all the videos and test cases are of sizes between (160 x120) to (320 x 240). Higher resolution impacts the speed of the system. The implemented model is a contributor to make it a system suitable for high quality videos too.

## 1.7 Thesis Outline

This section provides with the brief outline of the thesis.

**Chapter 2** provides literature review and all the techniques that are in practice related to the application has been discussed. This chapter will also provide with the related works in the field of surveillance, abnormal behavior detection, foreground estimation techniques, optical flow techniques and neural network implementations.

**Chapter 3** provides the details of implementation of optical flow. All the techniques used for optical flow implementation has been discussed and there models have been explained. Detailed mathematical models have been described for Horn-Schunk and Lucas-Kanade methods. Pros and Cons of each method will be discussed too.

**Chapter** 4 is an introduction the foreground extraction technique and its shortcomings. It further entails the implementation of Gaussian Mixture Model into the foreground extraction model for removing the effect of non-motion objects.

**Chapter 5** is dedicated to neural networks. It describes in detail all the techniques and parameters used within neural networks and explain the whole model for implementation. Applications of neural networks will also be discussed and overview of the functions used for neural network implementation will also be provided.

**Chapter 6** defines the proposed methodology for the application. It contains the main flow diagrams and a brief overview of how the model is actually implemented. It also describes the working environment, tools used and the specifications of resources used for this thesis.

**Chapter 7** provides with the implementation of the thesis. This chapter will first explain the algorithm of the thesis. Furthermore, this chapter discusses the implemented techniques one by one with the parameters used and provide with the snapshots of all the input videos used for the implementation.

**Chapter 8** is dedicated to the results produced by the implementation of all the techniques. It shows results produces by optical flow alone, foreground detection with GMM, optical flow results using foreground extraction wit GMM and finally the results from feed forward neural network.

**Chapter 9** is about the conclusions and future work. In this chapter, first the main aims will be reviewed and then the results of all the techniques will be concluded one by one.

## 1.8 Summary

In the above chapter, the research problem is discussed in detail along with the solutions to the problems. The chapter also provides the overview and explanation of the sections of prototype. Every section is discussed with the list of variable techniques. The applications of research are also being discussed. In the last section, the introduction to the implementation of prototype is discussed.

# Chapter 2: Literature Review

This chapter provides the detailed background of the research conducted and discussions are made on all the techniques used in this project. The chapter is divided into sections that provides with the brief introduction of the organization of prototype model. The main flow of data is depicted in Figure 2.1.

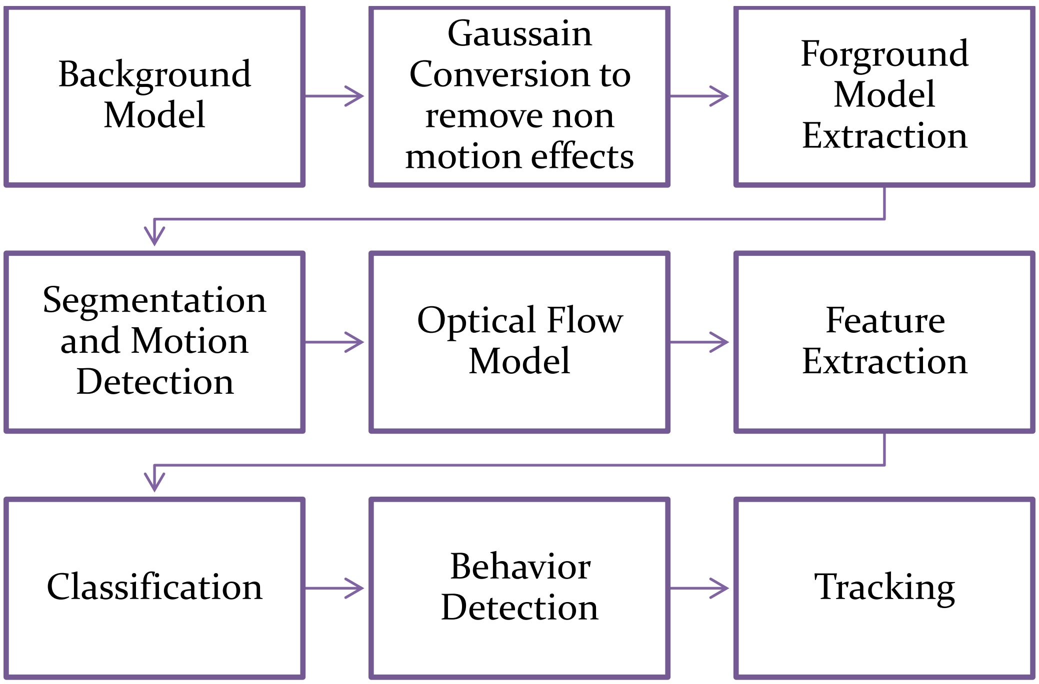



Figure 2.1: Data Flow of the Implemented Model

As the flow of data indicates, the prototype is a buildup of image processing, video processing, and pattern recognition and classification techniques. The following sections provide a description of all the blocks mentioned above.

## 2.1 Background

In the past few years, technology has become so much advanced. From home appliances to hi-fi military applications, things are becoming so easy to use and accurate. All the fields of science that had been in use are now combined to produce marvelous creations. In all this, the most emergent field during the past few years is **Computer Vision**. This field is a mixture of Control

Robotics, AI, and Machine Learning along with the essence of all other fields as shown in Figure 2.2.

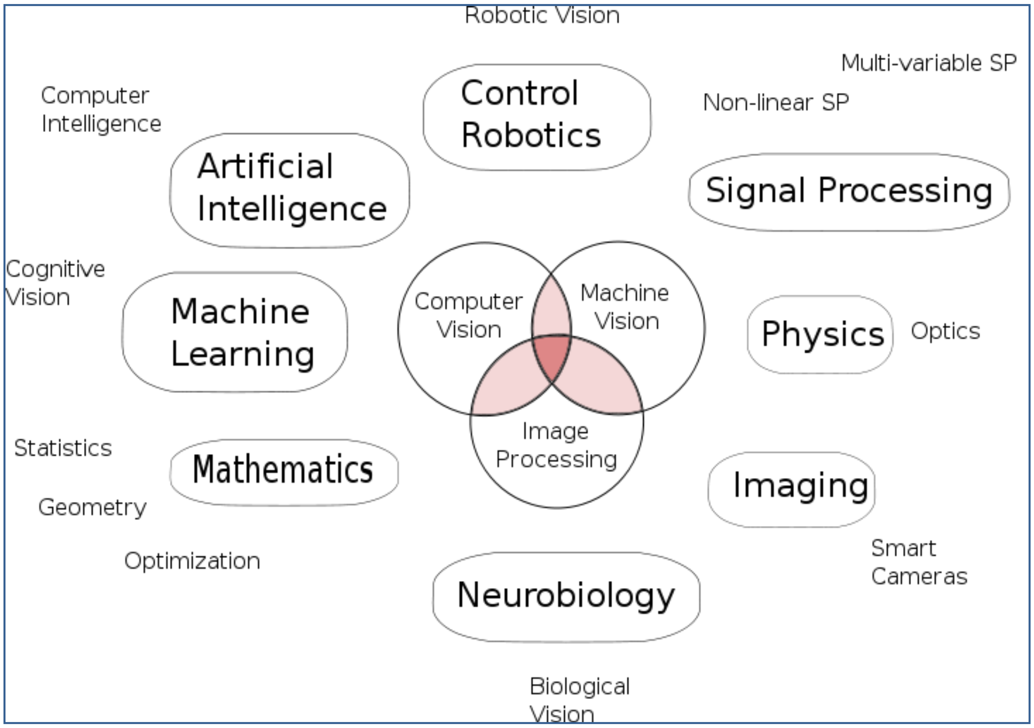



Figure 2.2: Relation of Computer Vision with other Fields

This field is equipped with methods for acquiring images and videos, processing their data, analyzing it to perform different functions, apply different techniques of tracking, recognizing, matching, and detecting abnormal behaviors within the data and much more. This field mainly aims to solve problems of real world and to perform numerical and symbolic operations on them to make decisions [**1**], [**2**]. This field is dedicated to create intelligent applications that imitate human behavior. MATLAB has a complete Computer Vision System Toolbox and Neural Network Toolbox dedicated entirely to this machine learning and AI [**3**]. These two toolboxes provide the user with numerous options to perform automating of an application and make it behave like human mind. Applications of Computer Vision range widely. The techniques may be applied to create a robot that takes care of elderly people at old age homes or they may be applied at homes to monitor suspicious activities outside the home and to prevent burglary. The field may find it's applications in banks in many forms. For instance, a machine that can count the money just like a human do, an automatic lock system that provides access only to authorized personnel or a surveillance system that rings an alarm on any activity that is not permitted. Another application is in air planes where the plane can be set to auto pilot in case of absence of pilot. There are thousands of examples on record that are benefitting mankind originating from this field.

Computer vision is a field that covers the basic transformations of automated image analysis being used in many areas. Robot Guidance, Controlling Process and automatic inspection are all applications that come in the domain of Machine Vision [**4**]. It is the process of combining automated image analysis with other methods to provide above applications.

Some examples of computer vision systems are given below:

- Industrial robots that are part of controlling processes
- Automated vehicles that are helpful for navigation
- Surveillance to detect abnormal behavior, counting people and monitoring etc
- Medical imaging applications to detect diseases and modeling the environments

The scope of this thesis is aimed towards surveillance in a way that uses minimal amount of data and classifies the video patterns. The target was to find a model that displays only motion vectors and out of these motion vectors, it produces the abnormal behavior detection results. Lots of work has been carried out on motion vectors and optical flow. Motion vectors have been used widely for Motion Estimation from an MPEG Stream or MPEG Compression Techniques [**5**]. Lately motion vectors have found their application in many fields like they can be used in motion blur, 3D Blur, Time Warping, Pixel Spread. Motion vectors are basically used for estimation of transformation, direction and speed for a certain frame. This phenomenon is called Motion Estimation. Motion vectors provide the user with the information about magnitude of displacement, velocity, direction and orientation between two frames. It is a key index to detect that a change in motion is being observed and is being tracked by use of motion vectors. By this change in motion from adjacent frames, the abnormal behavior can be detected.

## 2.2 Related Work

In surveillance or monitoring, different patterns are observed at different times. This variation is what we are focused towards [**6**]. We have examined work related to crowd behavior estimation, optical flow, foreground estimation, Gaussian Mixture Models and neural networks.

For real time object tracking, in [**7**], a method is proposed for real time tracking in which CamShift approach is used. An adaptive local search method is used that search for the best candidate object, to reduce the error rate and to prevent misclassification by surroundings. Furthermore, they have used kalman filter for prediction of object movement. Tracking is also accomplished by a technique named as contour based tracking [**8**], [**9**]. In [**8**], different types of contour based tracking have been explored. For contour based tracking, optical flow technique is used for initialization of contour. Markov random field theory is used for color based contour evolution. Crowd behavior is explored using lagrangian particle trajectories [**10**]. The main model starts with optical flow. Representative trajectories for the flow of crowd are obtained by clustering the particle trajectories obtained from the optical flow. Furthermore, chaotic dynamics are extracted using maximal Lyapunov exponent and correlation dimension. Finally an anomaly localization algorithm is fabricated to determine the size and exact position of the anomaly. Similar work is done in [**11**] in which the work is extended based on low rank optimization and decomposing the trajectories to capture the characteristics of trajectories.

Optical flow estimation is explored since 1940's [**12**]. According to J.J. Gibson, optical flow is the result of observing the pattern of light on the retina. In [**13**], the most famous method of optical flow has been introduced. This method is called Horn-Schunck (HS) in which the smoothness is assumed to be over the whole image. In comparison to HS method, Bruce D. Lucas and Takeo Kanade proposed a technique for computing optical flow that divides the image in portions and assumes smoothness over the neighborhood pixels [**14**]. This technique is since then, the most popular technique named as Lucas-Kanade (LK) approach. Optical flow is basically the 2D estimation of 3D motion of objects. Pattern of apparent motion of objects is termed as optical flow. A resultant vector field is computed that contains the horizontal and vertical components of each pixel that is termed as motion vector. Lots of work has been conducted in this field. Stefan and Michael present their work that is based on the spatial statistics of optical flow [**15**]. Here a large database is created using optical flow statistics from multiple videos and machine learning methods are deployed for markov random field model of the optical flow. Optical flow (LK) is used in combination with multi-scale Harris corner detection based on wavelet for vehicle tracking [**16**]. High frequency feature based on optical flow and multi scale histogram of optical flow has been used for anomaly detection in crowded scenes in [**17**].

In [**18**], optical flow is computed using a color background subtraction model that is base on spatially global Gaussian Mixture. Similar work is done in [**19**] based on LK algorithm and Gaussian background model. A comparison of pattern recognition and classification techniques is made in [**20**]. Fuzzy logic, neural networks, Markov random filed, support vector machines (SVM) and multi class SVM have been discussed and compared. Neural networks are fast, robust and capable of making complex decisions that are required for abnormal behavior detection. Multi object tracking is done using feed forward neural networks [**21**]. For non linear and non stationary features, a model is introduced in [**22**] where a multi layered feed forward neural network is used for supervised training.

## 2.3 Video Surveillance

The term surveillance refers to monitoring a certain area to avoid any unwanted activity or to record the results. The main use of surveillance is in protection and security. Surveillance has positive as well as negative effects associated with it. It is used at banks, offices, shopping malls, schools, colleges, sensitive areas and many other places to stop unwanted interference. Surveillance is not only limited to video monitoring. It is also used in many other means of communication like network security, telephones, biometric security check etc to ensure the protection of data. There are different types of surveillance. These are

- Biometric features like DNA, fingerprints, facial features, emotions, body temperature are all the monitored to check the situation at an area of any person
- Computer Surveillance that includes monitoring of data and traffic on internet [**23**]
- Social networking sites like Facebook, MySpace, and Twitter as well as many other communicating sites traffic is being monitored to protect from attackers.
- Surveillance cameras are used to visually monitor the activities within an area

- Data mining and profiling is done to observe any suspicious relationships between the data. It is mainly used in the protection of credit cards and online transactions.
- Telephones are being monitored to detect any suspicious activity [**24**]
- Aerial surveillance is done in many areas, especially done by military to protect national security.

There are other types of surveillance as well. The main reason of discussing some of the techniques here is to provide with a brief introduction of different surveillance types. Surveillance is very useful for government organizations, military areas, banks etc. The threats can deliberately be reduced with the usage of CCTV for monitoring. In just 2 to 3 years, the concept of surveillance has been changed. It is now used as automated system to detect any abnormalities and suspicious behavior. This is the main focus of thesis. The surveillance system designed here is intelligent enough to differentiate between normal and abnormal events going around.

## 2.4 Motion analysis

Motion analysis is basically to read the behavior of motion so that further processing can be done on it. It is the key ingredient when we talk about behavior detection in any video sequence. The analyzed motion is being fed into a classifier that first trains the data and then recognizes it on basis of the patterns analyzed. There are several steps that are related to motion analysis or motion estimation in which a video is read into frames or images, then operations on image sequences are performed to extract velocity estimate at each point in the image or scene or just selected points that are extracted by pre processing the sequences. The examples of motion analysis include:

**Egomotion** is a methodology in which 3D rigid motion is determined from an image sequence that is produced using the camera. [**25**]

**Tracking** is basically following a set of points, objects or blobs in the image sequence to determine the path of flow.

**Optical flow** is a technique that is used to determine the apparent motion for each point in the image that is basically that how that point is moving relative to the image plane. [**26**]

## 2.5 Classification of Motion Appearance

There are different patterns of movement of an object in a scene [**6**]. It is important to determine the type of motion to apply the algorithms as there are some types of motions that are specific to an algorithm. Then the motion of any object does not produce same results in every case. Orientation, speed, camera view are all associated with that trajectory and they effect the results as well. Not every type of motion can be entertained for the same algorithm. Some of the basic motion types that are tested in this thesis (Synthetic videos) and have produced amicable results are provided in the next sub sections.

### 2.5.1 Pan Motion

When an object in a video sequence moves linearly from left to right or from right to left, then this type of motion is considered pan motion. In this type of motion, the resultant motion should mainly be in horizontal direction and minor or no motion should be in vertical direction.

### 2.5.2 Tilt Motion

When an object in a video sequence moves either upwards or downwards, then this type of motion is considered tilt motion. In this type of motion, the resultant motion should mainly be in vertical direction and minor or no motion should be in horizontal direction.

### 2.5.3 Rotary Motion

Rotary motion is the motion of spinning and turning of an object that is around an axis. This motion has its results concentrated more on edges of the moving object. Especially when the moving object has a constant color or smooth surface, this result becomes absolutely true.

Objects in the video sequences for real videos are combination of all the above motions.

## 2.6 Foreground Extraction Models

Foreground extraction is basically the segmentation of background from foreground. The definition of background and foreground varies. In some techniques, background is considered as the part of scene that is static for all the frames and rest of the things are moving. In some cases, background is the part of an image or scene, that is considered to be static and the rest of the things that can be considered as moving are put in foreground. There are some cases as well in which the background is also changing according to the scene. It is basically a time constrained system in which, some objects in the foreground are considered as background if they stay static for a certain period of time. As soon as they start moving again, they become part of foreground again. The crudest form of finding the background model is to detect the moving objects by the difference between the reference frame and current frame where the reference frame is the background model [**27**]. This method is known as "frame differencing method".

There are mainly two types of techniques for background modeling (BM), ***adaptive*** and ***non-adaptive***. These two categories are explained below.

1. **Non-Adaptive Models:**

Non-adaptive techniques are basically those in which the background is defined once and it cannot be altered. The objects that are moved into background once will remain in background and the objects that are moved into foreground region will remain in foreground, no matter what happens. This technique is prone to errors when the environment has constant illumination changes or movements like moving trees in background, clouds appearance etc. In this case there is a need for re initialization of the BM or introducing a complex BM. These techniques will be suitable for simple and short term videos.

2. **Adaptive Models:**

In adaptive BM's, the background and the foreground are changed accordingly. This technique is mainly used at scenarios where the objects are complex and the background and foreground is constantly changing. The BM is initialized at the start of system but is adjusted according to the changing environment. This technique is computationally expensive as compared to non-adaptive models and hence should only be used in complex scenarios.

### 2.6.1 Techniques Overview

Different background/foreground extraction techniques are being developed. Each technique has its own pros and cons. Some of the techniques are described briefly in the following sections.

#### 2.6.1.1 Frame Differencing

It is the simplest approach in which the background is estimated at time t and the estimated background is subtracted from the input frame. By this the absolute difference is obtained which is then compared with the threshold to obtain the required foreground mask. Here the background that is estimated is just the previous frame in the sequence. The simple equation can be written for frame as:

$$|frame_i - frame_{i-1}| > Threshold \tag{2.1}$$

#### 2.6.1.2 Median Filtering

In this technique, the background is estimated by defining median at each pixel location in all the frames. This technique assumes that for more than half of the frames in buffer, pixel stays in the background. If we assume that there are more chances of background to appear in a scene, the median of the previous n frames can be used as a model. It is explained as:

$$B(x, y, t) = median\{I(x, y, t - i)\} \tag{2.2}$$

$$|I(x, y, t) - median\{I(x, y, t - i)\}| > Th \ (i \in \{0, \dots, n - 1\}) \tag{2.3}$$

#### 2.6.1.3 Mean Filtering

Mean filtering is done in the same as median filtering except that here the mean at each pixel location is used for background estimation.

$$B(x, y, t) = \frac{1}{n}\sum_{i=0}^{n-1} I(x, y, t - i) \tag{2.4}$$

$$\left| I(x, y, t) - \frac{1}{n}\sum_{i=0}^{n-1} I(x, y, t - i) \right| > Th \tag{2.5}$$

#### 2.6.1.4 Mixture of Gaussians

To track multiple Gaussian distributions, mixture of Gaussians is used. This technique is used widely to detect moving objects using static cameras [**28**]. Gaussian distribution is a continuous probability distribution in probability. It is defined by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.6}$$

The parameter $\mu$ is mean of the distribution. $\sigma$ is the standard deviation. Gaussian mixture models (GMM) refer to using the Gaussian distribution for multiple background distributions and refining the best out of them. In this method, density function for each pixel is maintained. It is designed to handle multi-modal background distributions. Parameters in this model are adaptively updated as this model is parametric and GMM is adaptive. Numerically the probability provided in a mixture of K Gaussians is explained as:

$$p(x) = \sum_{j=1}^{K} w_j . N(x|\mu_j, \Sigma_j) \tag{2.7}$$

In the above equation, $K$ is number of mixture components, $N$ is no of observations, $x$ is observation, $\mu_j$ is mean of $j$ component and $w_j$ is the weight or prior probability of the $j^{th}$ Gaussian defined as

$$\sum_{i=1}^{K} w_j = 1 \qquad \text{and} \quad 0 \leq w_j \leq 1$$

## 2.7 Vision Algorithms for Tracking

There are many vision algorithms that are designed for tracking. Addressing every algorithm is beyond the scope of this thesis. Some popular algorithms that had been considered for research are discussed briefly as followed.

### 2.7.1 Threshold Tracking

Threshold tracking is one of the simplest and easiest algorithms to implement. This algorithm can work on the frame differencing method that is described in section 2.5.1.1 or it can be accomplished with the help of connected components in the region that are sorted out after being through the range of thresholding. The principle of threshold tracking lies in the selection of a pixel intensity range,$T_{Range} = [T_{min}, T_{max}]$. For the particular cases where the threshold can remain constant within the specified range, the flow algorithm will be:

1. Define $T_{Range}$

2. Threshold the image / frame

   $I_{Thresh} = (I > T_{min})\&\& (I < T_{max})$

3. Find the connected components

   $C = ConnectedComponents(I_{Thresh})$

4. Filter the connected components and find the best match

$$M_{Best} = Filter(C)$$

The drawback of this technique is that when the background pixels range overlaps the foreground pixels range, the algorithm does not work. The pixel intensity cannot change suddenly for this algorithm which means that does not work well for sudden changes in the scene that is the key in identification of chaotic behavior. The basis of this technique is histogram extraction and it entirely fails when from the background clutter, histogram noise originates. There is another technique that is used against this and is based on K-means. The technique is Otsu Thresholding [**1**]

### 2.7.2 Particle Filters

Particle filters (PF) are amongst the most popular methods now a day. These filters are complex mathematical models and require lots of computations. They are relatively costly when it comes to hardware and mostly used for applications that need lots of computations in real time. Sequential Monte Carlo method (SMC) is another known term for particle filters which is a sophisticated model estimation technique based on simulation of the particles [**29**]. PF are usually used for Bayesian models estimation where similar to hidden Markov model (HMM) but typically they differ in state space model latent variables. These filters are based on randomly distributed particles that are contained within a frame bound. Movement of the particles can be as simple as the frame differencing method or as complex as needed for the real time applications like air planes target monitoring etc. There is weight associated with each particle that is determined by recording the particle's current position. The survival of the particles depends on the weight values. The particles higher in weight are more likely to survive while the particles with lower weights are prone to be filtered out and new particles with higher weights replace them. The algorithm for the basic steps of a particle filtering system is:

Particle filters are a popular way of performing target tracking and are used in real time application like jet crafts and other hard real time systems. Advantages and drawbacks, both are being discussed below:

**Advantages:**

1. For multiple targets in real time, particle filter are used.
2. The scope of particle filters can be extended for target acquisition.
3. Depending on the application, particle filters can be used as simple or more complex models as they have different choices for weights, re-sampling and motion.

**Disadvantages**

1. Lack of deterministic property leads to difficulty in testing.
2. Optimal number of particles is difficult to determine.
3. Computationally they are highly complex.
4. With the increasing dimensions of implementation model, number of particles also increase that leads to more consumption of resources.

### 2.7.3 Template Tracking

Template tracking is the technique for searching and tracking a template that is initially defined to be tracked within the bounds of an image or video in which a predefined template is being tracked within the image or video frames [**30**]. There is a vast number of applications of template tracking that includes robotics, used in manufacturing plant, used for car number plate recognition, use for facial tracking or as ways for detection of a target leaving or entering a premises. The basis of this technique lies in acquiring an image sample of the main target T (template) and searching for the best close match of the template within the next frames. The basic technique works well till the target is not changing dramatically from frame to frame. With few modifications in the original technique, the technique can be used very accurately to detect a single target for a very long time. The time required for this technique is more as compared to technique like optical flow.

In the very basic method, a convolution mask is used to detect the image template$T(x_t, y_t)$, where $(x_t, y_t)$ represent the coordinates of each pixel in the template, within the video frame or image$S(x, y)$, where $(x, y)$ are the coordinates of the pixels in the image to be searched for template. The template can be already defined or it can be selected dynamically. The technique is more easily implemented on gray images. The output of convolution will be highest based on the best match location.

Using Sum of Absolute Differences (SAD), template matching is used for problems that involve transition of frames. Pixel intensities are compared in this technique. The pixel that we have to find in the image has coordinates $(x_s, y_s)$ and intensity represented as $I_s\ (x_s, y_s)$ ad for the template, the template coordinates are $(x_t, y_t)$ and the intensity is $I_t\ (x_t, y_t)$ . From the above definitions, absolute difference in the pixel intensities can be defined as

$$Diff\ (x_s, y_s, x_t, y_t) = |\ I_s\ (x_s, y_s) - I_t\ (x_t, y_t)\ |. \tag{2.8}$$

$$SAD(x, y) = \sum_{i=0}^{T_{rows}} \sum_{j=0}^{T_{cols}} Diff(x + i, y + j, i, j) \tag{2.9}$$

Mathematical representation of basic concept which is constant search within every area of the search image to find a match for template T and to compute the SAD measure is provided in equation 2.10.

$$\sum_{x=0}^{S_{rows}} \sum_{y=0}^{S_{cols}} SAD(x, y) \tag{2.10}$$

Here rows and columns of the search image are denoted by $\mathbf{S_{rows}}$ and $\mathbf{S_{cols}}$ and for the template image rows and columns; it is denoted by $\mathbf{T_{rows}}$ and $\mathbf{T_{cols}}$. SAD score determines the best fit position for the template. The method stated above is simple when it comes to implementation but slowest when it comes to processing speed.

### 2.7.4 Kalman Tracking

The Kalman filter works on the basis of observing a series of measurements over time that contain noise and other deficiencies and in the end, provides with an estimate of the unknown variables that are more precise and accurate as compared to single measurement. The mode of operation for the kalman filter is recursive, through which the noisy data input is passed for the optimal estimate of the statistical model. The two main steps of implementation of kalman filter are:

1. Prediction of the target state

2. The prediction from step 1 and the velocity model output is used for prediction position update.

The optimal output of the kalman filter is estimated by the fact that for all the Gaussian noises within the model, kalman filter application tends to minimize the MSE of the results. Due to this property, this filter is popular for estimation of error and correct prediction of the state of model. It is also use for online real time tracking. The state of the system is predicted using a weighted average if the statistics. The main usage of weights is to pick out the values with better estimated uncertainty. The weights are calculated by the covariance method for the estimation of the uncertainty in the prediction state. The result is a new state estimate that is a weighted average whose value that lies between the predicted and measured state value. This process is repeated iteratively for the new estimate and the covariance indicating the prediction in the next iteration. Kalman filter works recursively and only the last best prediction is needed rather than keeping the track of entire history for the new system state. The state model for kalman filter is shown in figure 2.3 below.

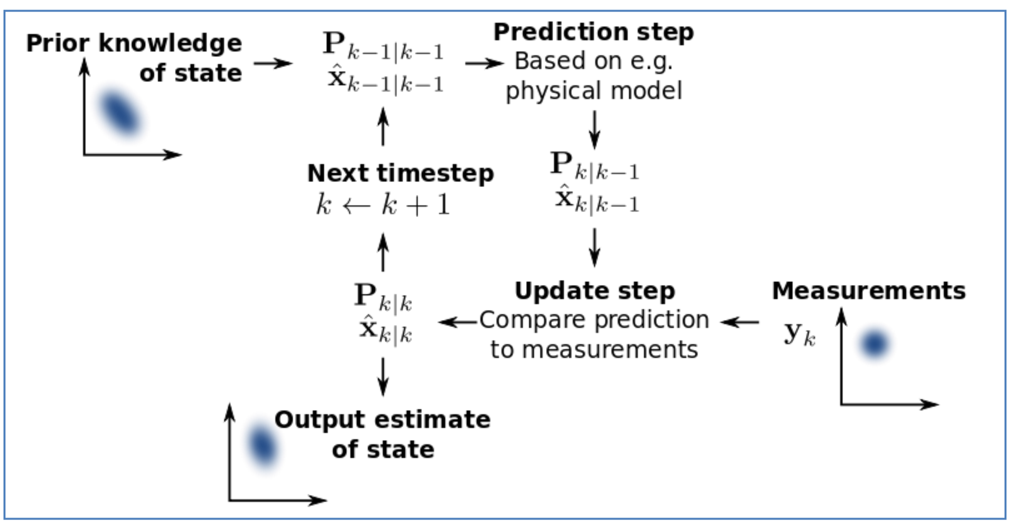



Figure 2.3: Kalman Filter basic concept [**31**]

### 2.7.5 Optical Flow Techniques

Optical flow is basically the 2D estimation of 3D motion of objects. Pattern of apparent motion of objects is termed as optical flow. A resultant vector field is computed that contains the horizontal and vertical components of each pixel that is termed as motion vector. Lots of work has been conducted in this field [**32**]. In the past few years, optical flow is designated for fields of image processing and control systems that includes segmentation of objects, application of motion detection, tracking, abnormal behavior detection etc [**33**].

#### 2.7.5.1 Methods to determine optical flow

Different methods have been devised to determine optical flow. The methods are listed below.

1. **Phase Correlation**: Phase correlation is the method based on frequency domain approach that is used estimate a relative translational effect between two similar images or video frames.

2. **Block Based Methods:** Block based methods are the methods where a block of a certain size or window is chosen and optical flow is applied on every block. The block is being matched in the next frame for tracking. The basis of this technique is minimization of SAD and maximizing cross correlation.

3. **Differential Methods:** Differential methods are the methods where optical flow is computed using partial derivatives of the horizontal and vertical components. This method has two applied techniques which are:

    a. *Lucas–Kanade method:* This method assumes consistency of flow in the local neighborhoods of the pixels within the predefined window search area and uses least squares criterion to solve the basic optical flow equations for all the pixels within that neighborhood area [**14**].

    b. *Horn–Schunck method :* This algorithm makes the assumption of smoothness in flow over the whole image. For the said assumption, this method prefers solutions that show more smoothness and aims to minimize the distortions produced in the flow [**13**].

c. *Buxton–Buxton method :* This method is edge based method that computes the flow based on the results of edges produced by motion in the image or frame.

d. *Black–Jepson method* : This method aims to find coarse optical flow using correlation methods [**34**].

## 2.8 Pattern Recognition Techniques

Assignment of a label to the provided input values is what is known as pattern recognition. Pattern recognition is a very important branch of machine learning. This field corresponds to the identification of a pattern of data input values and recognizes them as one of the assigned class. Classification is an example of pattern recognition in which the input values are mapped to matched assigned classes. These algorithms attempt to recognize all the inputs and assign them too their relative classes based on their statistics.

There are different categories of pattern recognition that depends on the type of network and the type of learning method used. Supervised, unsupervised and semi-supervised learning is used. In *supervised learning*, it is assumed that a whole data set for learning has been provided, containing all the input instances and all the matching targets that have the correct output to produce the desired results and accurate classification of data. In *unsupervised learning*, the data for training is not labeled before and there are different trials on the provided data sets to find the matching patterns and produce the desired output. Just recently, a technique that is known as *semi-supervised* learning is being explored that counts for a combination of both the supervised and unsupervised techniques. This uses both the labeled and unlabeled data.

### 2.8.1 Applications

Pattern recognition techniques are used in the field of medical. There are many devices in the modern day that aims to save lives and are build using pattern recognition techniques. These systems are called computer-aided diagnosis (CAD) systems. These systems help doctors to diagnose diseases as well as perform tests. The systems are used for micro level inspection that includes viruses in blood to diagnosis of broken bones in the body. Pattern Recognition is also used for applications like automatic speech recognition for security purposes, facial detection, monitoring of human actions and matching them with the approved actions and optical character recognition [**35**].

In image processing, the applications of neural networks are very wide. Examples of these applications include license plate recognition, face detection, medical diagnosis, fingerprint analysis, navigation and guidance systems for aircrafts and automated cars, automatic human behavior recognition, detection of chaos at some place etc.

### 2.8.2 Pattern Recognition Techniques

There are different pattern recognition techniques that have been explored for years. Some of these techniques are stated below that are categorized further as

1. **Conditional random fields (CRFs)**

A supervised learning method, conditional random fields of CRFs are statistical models that are applied in pattern recognition and machine learning for structural prediction. They are mainly used in applications for gene finding, object recognition and segmenting images. They are often used for the labeling of sequential input data that is more of natural like biological sequences, matching patterns of natural languages etc.

2. **Hidden Markov models (HMMs)**

A Hidden Markov Model (HMM) is a statistical model that has hidden states within the network model. This model is the simplest Bayesian network that is dynamic. HMM can either be supervised or unsupervised and are a generalization of a mixture model in which the hidden variables are related through Markov process rather than being operated independently.

3. **Decision trees**

In application like decision analysis, the technique known as decision trees is used. It uses a tree like graph that models the sequence on the possible outcomes and consequences. They are commonly used in application like research analysis, decision making so that it is easy to eliminate all the possible outcomes and a better way of understanding the whole possibility set is developed. A decision tree is a non parametric technique.

4. **K-nearest-neighbor algorithms**

Another technique in implementation is k-nearest neighbor algorithm in which k is a user defined value that is constant and is used to specify the area for neighbors. In pattern recognition, k-nearest neighbor algorithm (k-NN) is a non-parametric method for the classification of objects subject to the closest training patterns in the feature space. This algorithm is the simples of all machine learning algorithms. The classification of an object is dependent on the majority vote of the neighbors and the object is assigned to the class that has the most common attributes as of its neighbors.

**5. Neural Networks**

Neural networks are considered to be amongst the most sophisticated and reliable pattern recognition techniques. They are basically a replica of human brain and are made intelligent to take decisions like a human brain. The neural network or artificial neural network (ANN) is composed of artificial neurons and these neurons transmit and compute the decisions base on inputs. They are used for solving artificial intelligence problems and are highly recommended in robotic applications. It is a network that consists of a group of interconnected natural or artificial neurons that are being subject to a mathematical or complex computation model for processing information that is based on a connectionist approach to computation [**36**]. In most applications, ANNs are adaptive and change their structure based on the conditions and needs of the system.

## 2.9 Summary

This chapter presented the literature review of the thesis. It has started with the parent field i.e. computer vision and in the following sections, rest of the study has been discussed. Related work that is being done with respect to each field is presented. In the next sections, all the techniques that are implemented have been discussed and different models for each technique have been stated. It has provided in details the techniques that are used for each method. In the last section, different techniques used for pattern recognition has been discussed.

# Chapter 3: Optical Flow

## 3.1 Introduction to optical flow

Optical flow is the pattern of apparent motion of the objects velocities within an image or video frame [**32**]. It is an approximation of the local image that is based on the derivatives in the given image or video frame sequence. It basically specifies about the translation of pixels between adjacent images. Optical flow can be the result of the relative motion between the object of interest and the observer. Eventually optical flow is used for the information about the spatial details of the objects of interest and the rate of change of the details.

As already stated, optical flow estimation is in research since 1940's [**12**]. According to J.J. Gibson theory, optical flow is the result of observing the pattern of light on the retina. In [**13**], one of the most famous method of optical flow, that is still in practice and has provided base for many comparative studies, has been introduced. This method is called Horn-Schunck (HS) in which global smoothness is assumed for the whole image. In comparison to HS method, Bruce D. Lucas and Takeo Kanade proposed a technique for computing optical flow that divides the image in portions and assumes smoothness over the neighborhood pixels [**14**]. Fleet and Weiss provide a tutorial introduction to gradient based optical flow [**37**]. John L. Barron, David J. Fleet, and Steven Beauchemin [**34**] provide a performance analysis of a number of optical flow techniques.

Optical flow method is the result of relative motion between observers and objects. In this method, motion vectors are calculated that provides us with the information about the horizontal and vertical component of the pixel, speed of the pixel across the image and the direction in which it is moving.

## 3.2 Computation of Optical Flow

The basic implementation of optical flow starts with the motion constraint equation. In an image or video frame, each pixel is represented by an intensity value. Each pixel; has a projection of it that correspond to space of image onto the image plane. When the object moves in the plane, the projection of the object also moves along. Optical flow is the motion vector that shows the direction, velocity and magnitude change from image to image for the change.

Optical flow is directly related to the object of observation. If we increase the speed of object, optical flow speed will also be increased.

**(a)** **(b)**

Figure 3.1: a) Original Video b) Optical Flow representation of (a)

From the above figure it can be seen that optical flow is clearly obtained for the object of interest that is in motion.

## 3.3 Basic Assumptions for Computation of Optical Flow

There are different techniques used for computation of optical flow. For every technique, optical flow calculation assumes the basic three assumptions.

1. **Brightness consistency:** It assumes that the brightness associated with the pixels in a small region within the image remains the same, although the location may vary.

2. **Spatial Coherence:** The points that are in neighborhood usually have the similar type of motion and surface.

3. **Temporal Persistence:** Temporal persistence states that the image motion of the frame sequence changes gradually over the period of time.

The methods that have been tested for this thesis are differential methods and are explained in coming sections. These methods are *Lucas–Kanade* method and *Horn–Schunck* method .

## 3.4 The 2D Motion Constraint Equation

In optical flow, the motion is calculated between two consecutive frames that are displaced by time difference δt and are operated at 't' and 't+δt' for every pixel location. HS and LK are called differential methods as their bass lies in the approximation of the image input based on Taylor series. In other words, they use partial derivatives with respect to the brightness consistency, temporal and spatial coordinates.

For the calculation, we first assume that I (x, y, t) is the central pixel in an n x n neighborhood of pixels and moves by δx, δy in time δt to I (x+δx, y +δy, t+δt). As I (x, y, t) and I (x + δx, y + δy, t + δt) are the image frames of the same point, we have:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{3.1}$$

This assumption is the basis of the 2D Motion Constraint Equation that is shown in Figure 3.2. This assumption in equation 3.1 is true for the first approximation with condition that δx, δy, δt are not very large in value. The 1$^{st}$ order Taylor series equation can be applied to the equation above to obtain equation 3.2.

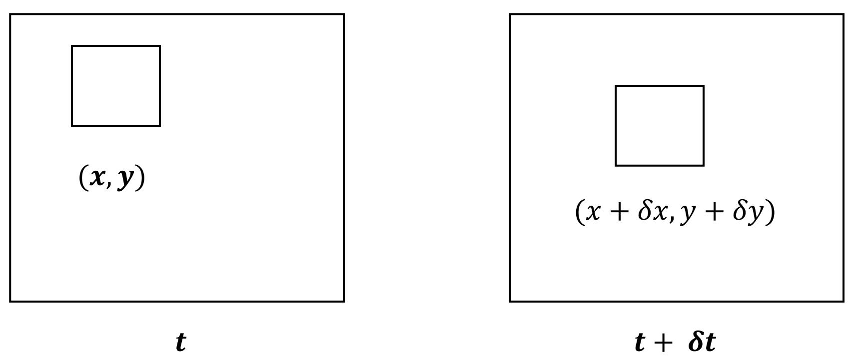



Figure 3.2: The image at (x, y, t) is the same as at (x + δx, y + δy, t + δt)

$$I(x + \delta x, y + \delta y, y + \delta t) = I(x, y, y) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + H.O.T \tag{3.2}$$

Where H.O.T means higher order terms, which are very small that they can be ignored.

From equation 3.1 and 3.2, it follows that:

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \tag{3.3}$$

or

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t}+\frac{\partial I}{\partial y}\frac{\delta y}{\delta t}+\frac{\partial I}{\partial t}\frac{\delta t}{\delta t} \tag{3.4}$$

That results in

$$\frac{\partial I}{\partial x}Vx+\frac{\partial I}{\partial y}Vy+\frac{\partial I}{\partial t}=0 \tag{3.5}$$

Where $V_x$,$V_y$ are the $x$ and $y$ components of the velocity or optical flow of $I(x,\ y,\ t)$. $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the partial derivatives of the image frame at position $(x,y,t)$ in the described directions. From above, $I_x$,$I_y$ and $I_t$ can be written for the partial derivatives as shown in equation 3.6 below:

$$I_xV_x + I_yV_y = -I_t \tag{3.6}$$

Equation 3.6 is the main 'optical flow constraint equation'. Equation 3.6 is an equation with two unknowns and cannot be solved directly. This phenomenon of uncertainty is known as *aperture problem* of the optical flow. To find the optical flow, more equations are needed that have some additional constraints and each one is defined by the optical flow technique. In order to solve the aperture problem, the additional constraints are expressed in the form of equations that are described for the two methods that have been tested for this thesis.

## 3.5 Horn–Schunck Method

The Horn–Schunck (HS) method for estimation of optical flow is a method in which we assume that there is smoothness all over the image. It's a global method that assumes the same neighborhood pixels for the whole frame. It aims to minimize the irregularities in optical flow and select solutions with more smoothness in output [**13**].

The flow is devised as the global energy functional which is then brought to minimization. This function is given for 2-D image streams as defined below:

$$E = \int\int \left[\left(I_xu + I_yv + I_t\right)^2 + \alpha^2 + (|\nabla u|^2 + |\nabla v|^2)\right] dxdy \tag{3.7}$$

Where $I_x$, $I_y$ and $I_t$ are the partial derivatives of the image intensity values along the horizontal, vertical and time dimensions respectively and parameter α is regularization constant. Larger values of α brings smoother flow. This energy functional can be minimized by solving the Euler–Lagrange equations. These equations are:

$$I_x\left(I_x u + I_y v + I_t\right) - \alpha^2 \Delta u = 0 \tag{3.8}$$

$$I_y\left(I_x u + I_y v + I_t\right) - \alpha^2 \Delta v = 0 \tag{3.9}$$

The above equations are linear in u and v and may be solved for each pixel in the image frame. However, as the solution of the constraints depends on the neighboring pixels values of the flow field, iteration is a necessary condition for updating. The following equations for iteration have been derived:

$$u^{k+1} = u^{-k} - \frac{I_x(I_x u^{-k} + I_y v^{-k} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{3.10}$$

$$v^{k+1} = v^{-k} - \frac{I_x(I_x u^{-k} + I_y v^{-k} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \tag{3.11}$$

Where k is the last calculated output and k+1 is the next iteration. The main advantage of this method lies in the high density of flow vectors however, the disadvantage is that it its sensitivity to noise than other local methods.

## 3.6 Lucas Kanade Method

This method is followed for this thesis. Lucas–Kanade (LK) method is also a differential method and is developed by Bruce D. Lucas and Takeo Kanade [**14**]. This method brings forward an additional term to the optical flow by this assumption that the flow is constant in a local neighborhood or the whole image frame is divided into smaller portions for obtaining the smoothness.

It is assumed that the flow ($V_x$,$V_y$) is constant in a small window of size mxm with m > 1, which is centered at Pixel x,y and numbering the pixels within as 1...n, n = m2, a set of equations can be found:

$$I_{xn} + V_x + I_{yn}V_y = -I_{tn} \tag{3.12}$$

With the above equation, two more equations are generated for the unknown values and the system is over-determined. Hence:

$$\begin{bmatrix} I_{x1} & I_{y1} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ \vdots \\ -I_{t2} \end{bmatrix} \tag{3.13}$$

or

$$A\vec{v} = -b \tag{3.14}$$

The main advantage of Lucas–Kanade algorithm is that it works very well in the presence of noise. However, the disadvantage is that it does not produce high density optical flow.

## 3.7 Limitations of Optical Flow

Optical flow has its own limitations. It can be zero when it comes to 3-D object in motion. An example of this is a rotating sphere with a perfectly uniform surface distribution. In this case, the apparent motion is zero due to the fact that the sphere’s image does not change with time. For motion homogeneous objects, optical flow is not recommended.

## 3.8 Summary

In the above chapter, different methods for optical flow and their implementation has been discussed. It also specifies the advantages and disadvantages of each technique. The methods tested for this thesis have been discussed. In the last, limitations of optical flow have also been discussed.

# Chapter 4: Foreground Detection

Foreground extraction is a technique in the fields of image processing and computer vision wherein an image's foreground is extracted for further processing. For a surveillance application, generally the image's regions of interest are objects (humans, cars, text etc.) in its foreground. Due to the fact that optical flow technique can take into account any motion statistics, the main model is first subject to the foreground detector system that mixes up with the Gaussian mixture model to remove the effect of background clutter and to extract the objects in motion only. Foreground detection alone cannot be used as it generates noise at some places and is not very effective to take up alone [**28**]. Gaussian Mixture Models are used widely to remove the noisy effects in any image or video processing algorithm. These two techniques are used in combination to produce the results that would be noise free and provide a very refined data to be processed by the optical flow model. Here noise is everything except the objects in motion, and extraction of useful information with minimum data. This is done by using foreground extraction with Gaussian mixture model [**38**] and then applying optical flow method onto the extracted model. [**26**]. By use of these techniques, noise is eliminated at the first level and then motion vectors are generated only for the objects in motion.

## 4.1 Principles of Foreground Detection

Many algorithms for foreground detection have been discussed and tested in [**39**]. The comparisons will reference in coming sections as well. In this paper they have compared different methods and based on the results, some points have been deduced. The five principles are explained as below:

1. In the first rule, semantic differentiation is discussed where it is stated that that merely the background maintenance module should not be referred for handling of objects. This is supported by the fact that there are some objects that may be misclassified as background object when it is a foreground object, or vice versa, then only this background maintenance module is not enough to accurately track.

2. The objects of interest should be segmented on their first appearance into the scene. This point is mainly use for adaptive techniques. Recognition and tracking both are very

complex tasks and require exact pre processing to be done accurately. These two will perform best when a good background subtraction technique is employed that focuses more on objects of interest.

3. There are cases when we have multimodal views. Some objects may appear to move but they must be part of background. This is where rule number three states about the requirement of criterion that defines stationary pixel level. This point is related to rule above where the criterion for object of interest should be defined. The pixels that satisfy this criterion, they are marked as background and the emphasis is transferred to the rest of objects that are actually foreground objects.

4. The fourth point is basically about adaptation in which the emphasis is on this thing that which objects goes into the background after how long. This criterion is basically a compromise between the optimal conditions. This rule states that the background model should be intelligent enough to identify that which objects are part of background now and which objects are now foreground objects. The system should be adaptable to both sudden and gradual changes in the background.

5. There are problems of sudden intensity changes in the scene. The model should be adaptable to these changes as well. It should also keep into account the changes that are due to differing spatial scales.

## 4.2 Schemes for Foreground Detection

Different schemes used for foreground extraction have been explained and compared in literature review. They are discussed in detail here. Here the video function is used as taking V(x,y,t) as a video sequence in which 't' is the time dimension of sequence and 'x' and 'y' are the pixel location variables.

### 4.2.1 Using frame differencing

As discussed before, frame differencing the basic method of foreground extraction in which the difference of consecutive two frames is calculated and foreground is the result. For the settings of *x,y* stated above, the equations are calculated as below.

$$D(t+1) = |V(x,y,t+1) - V(x,y,t)| \tag{4.1}$$

In the above equation, $D\ (t+1)$ is the frame difference at time $t$+1. The background here is the frame at time $t$ that is $V(x,y,t)$. Although it looks like that the background is removed, this approach is meant only for the cases when we have all the moving foreground pixels and all the background pixels are static. A threshold "Th" is used to improve the results of this subtraction.

$$|V(x,y,t) - V(x,y,t+1)| > Th \tag{4.2}$$

The difference image's pixels' intensities are thresholded or filtered on the basis of value of 'Th'. The accuracy of this approach is dependent on speed of movement in the scene. Faster movements may require higher thresholds.

### 4.2.2 Mean filter

In this technique, for the calculation of background image, the preceding frames in the sequence are averaged to find the mean. To calculate the background image at the instant $t$, we have the following equation:

$$B(x,y) = \frac{1}{N}\sum_{i=1}^{N} V(x,y,t-i) \tag{4.3}$$

Here $N$ is the number of preceding frames that are taken for averaging. $N$ is dependent on the video speed and the movement happening in the video [**39**]. After calculating the background $B(x,y)$, it can then be subtracted from the video frame $V(x,y,t)$ at time $t$=t and threshold it. Thus the foreground is

$$|V(x,y,t) - B(x,y)| > Th \tag{4.4}$$

Where $Th$ is threshold. Similarly median can also be used instead of mean in the above calculation of $B(x,y)$ for median filtering.

### 4.2.3 Gaussian mixture models

A Gaussian Mixture Model (GMM) is a parametric model that is represented in the form of probability density function defined as a weighted sum of Gaussian component densities. In [**18**], optical flow is computed using a color background subtraction model that is base on spatially global Gaussian Mixture Models. A Gaussian mixture distribution is defined as a distribution in which a multivariate distribution exists that comprises of a mixture of one or more multivariate Gaussian distribution components. For every multivariate Gaussian component, its mean and covariance is defined, and the whole mixture is defined by a vector of mixing proportions from all the distributions. In this technique, it is assumed that every pixel's intensity values in the video can be modeled using a Gaussian mixture model. A simple model determines the place of foreground and background intensities. The pixels which do not fulfill the criteria of background pixels are placed in the foreground pixels array. At any time t, a particular pixel $(x_0, y_0)$ history is stated as:

$$X_1, \ldots, X_t = \{V(x_0, y_0, i) : 1 \leq i \leq t\} \tag{4.7}$$

This whole history is mapped by using a mixture of K Gaussian distributions:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} N(X_t | \mu_{i,t}, \Sigma_{i,t}) \tag{4.8}$$

Where

$$N(X_t | \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{\left|\Sigma_{i,t}\right|^{1/2}} \exp\left(-\frac{1}{2}\left(X_t - \mu_{i,t}\right)^T \Sigma_{i,t}^{-1}(X_t - \mu_{i,t})\right)$$

For the above technique, an on-line K-means approximation is used for updating Gaussians. Numerous improvements of this original method have been developed by Stauffer and Grimson [**40**] have been proposed in the reference and a complete survey of the techniques and there comparisons can be found in Bouwmans et al. [**41**]

## 4.3 Summary

This chapter presented some basic models for foreground extraction. It has provided with the detailed implementation of each model. In the last section, the technique, that has been used for this thesis, is stated and explained. Mathematical model for the technique has also been devised.

# Chapter 5: Neural Networks

The main concept of neural network is basically inspired by human brain or the biological nervous system. Human brain is the ultimate target of Artificial Intelligence since many years. Scientists have been trying for years to produce intelligent machine that perceive and respond exactly like humans. The early theoretical base for neural networks was proposed by Alexander Bain [**42**] in 1873 and William James [**43**] in 1890. Both of them stated through their work that thoughts and body activity is a result of interactions among neurons within the brain. Neural networks are chosen mainly as they are very good for complex decision boundary problems over many variables.

## 5.1 Introduction to Neural Network Theories

In Bain's theory [**42**], a certain number of neurons are fired when any activity takes place. When these activities repeat themselves, the connections between these neurons become stronger. According to him, this repetition leads to memory formation.

James's theory [**43**]is quite similar to Bain's theory. However, according to him, memories and actions are the results of electrical currents that are flowing among the neurons within the brain. In his model, no individual connections for memory or actions are required as the main focus is on the flow of electrical currents.

Since many years, we are seeing robots and machines performing human actions. However, they lacked the key ingredient of human brain, the emotions and the reflex actions. Neural network is a step towards this weakness and is configuring systems in a way that is near to human reception. In human brain there are millions of functions performed by small neurons. Human brain has the capability to do multi tasking. By taking this thing into account, neural networks have been designed for effective multitasking and solving real time systems that are not easy to solve with other methods.

## 5.2 Applications of Neural Network

The main functions performed by neural networks are function approximation, Classification and Data Processing.

- Classification: Pattern recognition, feature extraction, image matching
- Noise Reduction: Recognize patterns in the inputs and produce noiseless outputs
- Prediction: Extrapolation based on historical data Pattern and sequence recognition

Different applications of neural network are described in Table 5.1 below

Table 5.1: Applications of Neural Networks

| Industry | Application |
|---|---|
| **Aerospace** | High-performance aircraft autopilot operations, flight trajectory simulation, control systems of the aircraft, enhancements for the autopilot, simulation of the components of aircraft and detection of any fault in the aircraft |
| **Automotive** | Automatic guidance system for the automobile and activity analysis for warranty claims |
| **Banking** | ATM Card fraud detection, detecting outliers in transactions ,checking and evaluation credit card applications and viewing abnormal patterns of transactions for any account |
| **Defense** | Used in defense application for automatic weapon steering, target tracking, facial recognition, development of new kinds of sensors, object discrimination, radar and image signal processing including data compression, feature extraction and noise suppression, and signal/image identification |
| **Financial** | Loan advising applications, Real estate |

| | |
|---|---|
| | comparisons, screening of mortgage, bond rating for the corporate sector, tracking activities of credit card usage, corporate financial analysis and currency price prediction |
| **Insurance** | Evaluation of applications for policy and optimization of products |
| **Telecommunications** | Real time translation of the spoken language, Image and data compression for speech and video data, automated information services for customers, and automated customer payment processing systems |
| **Speech** | Speech recognition, speech compression, words and vowels classification, detection of different speeches and text-to-speech synthesis |
| **Securities** | Market surveillance for any chaotic event, analysis, automatic check on highways for detection of accidents, and automatic monitoring systems for airports etc |
| **Robotics** | Trajectory control for a moving robot, action control of robots by embedding AI and real time processing of events |
| **Oil and gas** | Exploration |
| **Medical** | Breast cancer cell analysis, EEG and ECG data analysis, optimization of transplant times, |

| | |
|---|---|
| | hospital expense reduction, hospital quality improvement, and emergency-room test advisement |

## 5.3 Categories for Learning

There are three types of categories for learning. They are supervised, unsupervised and semi-supervised. These three categories are described in detail below.

### 5.3.1 Supervised Neural Networks

These networks are trained to produce the desired output by 1st providing the network with pre-labeled data. Here the training data comprises of a data set of training examples where each example consists of a pair containing an input object (feature vector) and target value that is desired for that input feature. In this type of learning, the training data is analyzed to predict the output. These networks work best for dynamic systems. There are four types of supervised networks that exist in neural network toolbox of MATLAB:

- **Feed Forward Neural Networks:** One way connection from input to output layers, commonly used for pattern recognition, prediction and non-linear fitting of data.

- **Radial Basis Functions:** These networks are basically used as an alternative method for designing non-linear feed forward networks.

- **Dynamic Networks:** These networks are based on the usage of memory and dynamic feedback connections that are used for the pattern recognition of spatial and temporal data.

- **Learning Vector Quantization (LVQ)**: LVQ networks are used for the data pattern that are not separable linearly.

Out of these four, feed forward neural networks have been used for this thesis and will be explained the upcoming sections.

### 5.3.2 Unsupervised Neural Networks

In this type of network, no prior information about the output is provided to the network. The network trains the data, learns the similar patterns and based on that, classifies the output. It is often referred as the problem of finding the hidden patterns in the unlabeled data. As the data provided to the network is unlabelled, there is no error function generated. The network adjusts itself to find the optimal results. Two types of unsupervised techniques for neural networks are:

- **Competitive Layers:** In this type of network, group of similar input vectors is recognized and input are automatically sorted to categories. They are normally used for classification and pattern recognition.

- **Self Organizing Maps (SOM):** SOMs are same as competitive layers; however, they differ from them in the quality of topology preserving of the input vectors. They assign

### 5.3.3 Semi-Supervised Neural Networks

This type of learning makes use of both supervised and unsupervised learning to operate. It has labeled data (smaller set) as well as unlabelled data (larger set) for the classification and recognition of patterns of data. The methods that are use for this type of learning are:

- Generative models
- Low Density Separation
- Graph Based Methods
- Heuristic approaches

Explanation of these methods is beyond the scope of this thesis.

## 5.4 The Simple Model

These networks are composed of simple elements that are operating in parallel. As in nature, the connections between elements largely determine the network function. A neural network has different values of connections between the elements that affect the strength of the signal through this connection. These element connections are called weights.

A neural network can be trained by adjusting these weights. Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Figure 5.1 illustrates this. 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.

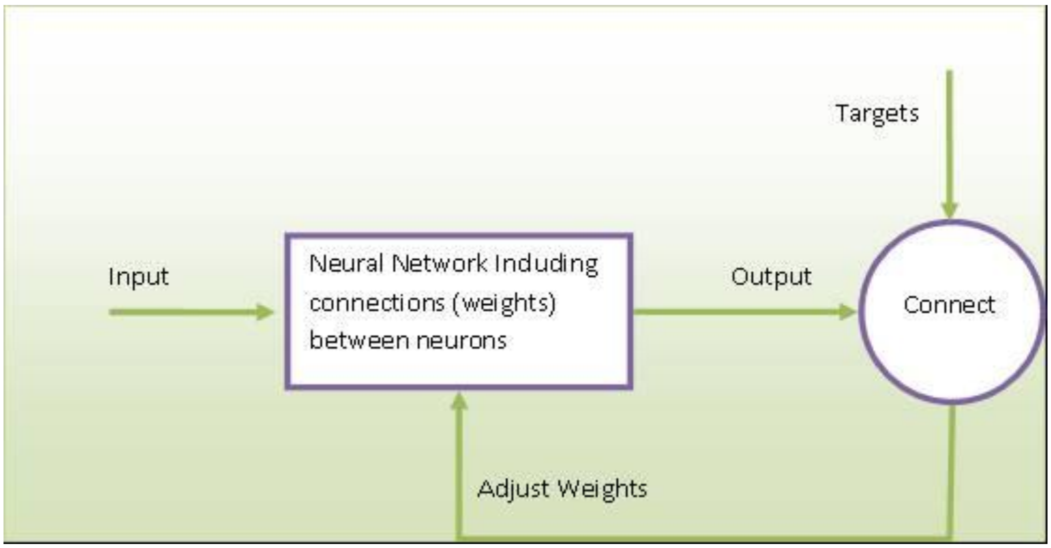



Figure 5.1: Neural Network Basic Model

Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, and 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. This chapter explains how to solve problems in function fitting, pattern recognition, clustering, and time series.

## 5.5 Preprocessing and Post-processing

In neural networks, some pre-processing is required for the input and target data before passing it to the network. By pre-processing, the networks efficiency is improved as the best features are selected after preprocessing that performs faster and necessary information is also not loosed. In the pre-processing methods, data is manipulated by scaling and normalizing the data. Principal components analysis is used for reducing dimensions of large data that is resulting in slow speed of network. Regression analysis is also used as a pre-processing technique.

Post-processing is basically used to visualize the results and analyze the network performance. The results are usually obtained in the form of confusion matrix, regression analysis, ROC Curves, MSE of the training, validation and testing performance and error histograms. The results are also calculated in the form of some standard parameters as true positives and negatives that are explained in the upcoming sections.

## 5.6 Working of Neural Networks

In neural networks, fixed number of inputs is assigned to both static and dynamic type of networks. Preprocessing is normally required as it defines the success of a classifier and helps the network to perform better. There are mainly three layers in the network. The first is an input layer; the $2^{nd}$ layer is the hidden layer and the third layer is the output layer. The important points that are needed to be considered for the neural network are recognition of presence of target data and extraction of the data from the input stream.

The network algorithm is:

1. Collection of data
2. Creation of the neural network
3. configuration of the network
4. Initialization of the weights and biases
5. Training of the network for the provided values
6. Validation of the network to test if the network is performing well
7. Simulating the network for new inputs or test data

The figure below shows the artificial model of a neuron.

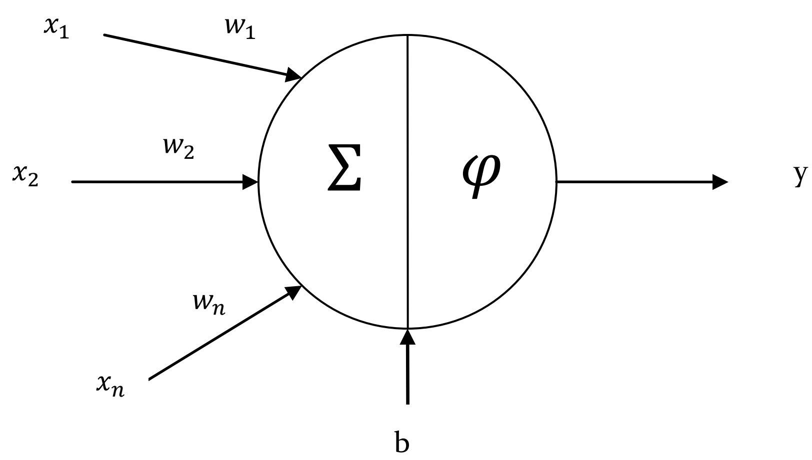



Figure 5.2: The artificial model of the neuron

The transfer function is

$$y = \varphi\left(\sum_{j=1}^{n} w_j x_{j-b}\right) \tag{5.1}$$

For the equation above, if $\alpha \to 0$, then $\varphi(z)$ is the *sign* function. For this special case, the mapping generated by the neuron is following:

$$y = sgn\left(\sum_{j=1}^{n} w_j x_j - b\right) \tag{5.2}$$

The formal proof of the statement above is beyond the scope of this thesis.

## 5.7 Feed Forward Neural Network

Feed forward neural networks are the simplest of neural networks. They have information flowing in just one direction, from input to output. These networks are used for pattern recognition and classification. They do not have any feedback element. Feed forward Neural Network is explained graphically in Figure 5.3. It shows one input layer with 3 input neurons, two neurons in the hidden layer and one neuron in the output.

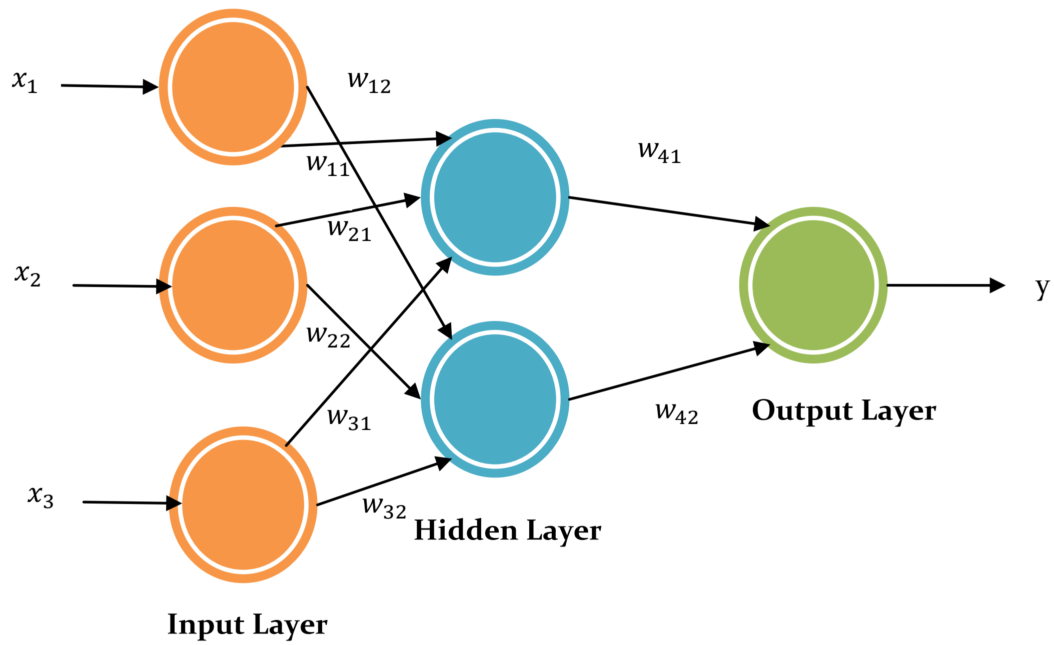



Figure 5.3: Three layer Feed Forward Network

Figure 5.3 also shows the weights associated with each connection. For every layer, a bias is added to quantify the output. Data always propagate in the direction of network inputs to the network outputs in the feed-forward networks. The activation function of a hidden layer unit is defined as:

$$O_c = h_{Hidden}\left(\sum_{p=1}^{P} i_{c.p}w_{c.p} + b_c\right) \tag{5.3}$$

Where

$$h_{Hidden}(x) = \frac{1}{1 + e^{-x}}$$

In equation above, $O_c$ is the current hidden layer unit c output, P can be anyone of the number of units in the previous hidden layer or number of inputs presented to the network. $i_{c.p}$ is the input unit c connected to the previous unit p. Similarly $w_{c.p}$ is the weight associated with hidden layer c to previous hidden layer or input p. $b_c$ is the bias for the layer.

In above equation, $h_{Hidden}(x)$ is the sigmoid activation function of the unit and is shown in Figure 5.4. There are different activation functions that exist but sigmoid was chosen for this application as it limits the values of minimum and maximum target value data.

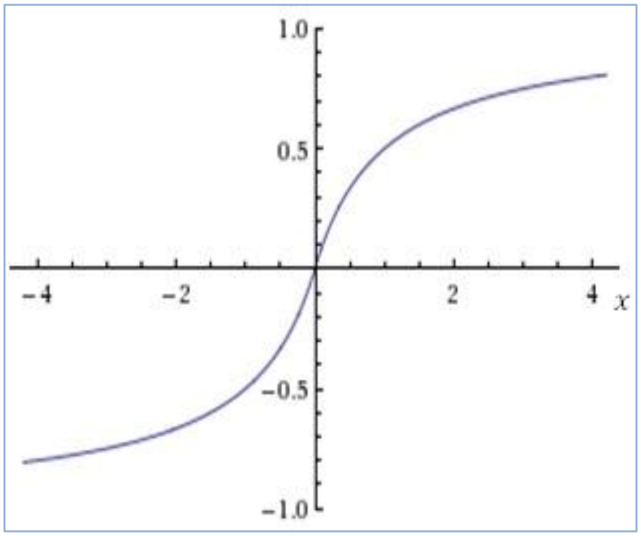



Figure 5.4: Sigmoid Activation Function

## 5.8 Performance Estimation Parameters

There are different performance parameters that are associated with neural networks. These parameters define the goodness of the results. The parameters are Overall MSE, Overall error percentage, sensitivity, specificity, true positive rate, true negative rate, Precision, Recall (Sensitivity), Specificity, F-measure and percentage wrong classified.

These parameters are explained as:

1. True Positive (TP): The number of values that are positive outcomes of one class
2. True Negative (TN): The number of values that are negative outcomes of other class
3. False Positive (FP): The number of values that are negative and falsely classified as positive
4. False Negative (FN): The number of values that are positive and falsely classified as negative

5. Sensitivity (Recall): It is defined as the proportion of actual positives in the output that are correctly identified. It is shown in form of percentage.

$$\text{Recall or sensitivity} = \frac{tp}{tp + fn}$$

6. Specificity: It is defined as the proportion of actual negatives in the output that are correctly identified. It is shown in form of percentage.

$$\text{True negative rate or specificity} = \frac{tn}{tn + fp}$$

7. Accuracy: It is defined as the percentage of correctly classified values. It is also calculated in the form of MSE.

$$\text{Accuracy} = \frac{tp + tn}{tp + fn + tn + fp}$$

8. Percentage Error: It is defined as the overall wrong classified values.

$$\text{Percentage Error} = 100 * \frac{fn + fp}{tp + fp + tn + fn}$$

9. F-Measure: It is a measure that produces a value that is the combination of both precision and recall and its value lies between '0' to '1'. If the value is towards zero, the network's performance is poor. If it's going towards 1, the network is performing well. The formula is defined as:

$$F = 2 * \frac{precision.recall}{precision + recall}$$

## 5.9 Summary

This chapter introduced the basic concepts of Neural Networks. It then explained all the applications of neural networks with respect to their area. The basic neuron model has been explained. Further, feed forward network has been discussed. In the last section, performance measures with formulas for the NN have been stated and explained.

# Chapter 6: Proposed Methodology

This chapter is about the main research methodology used in this thesis and also provides an insight on the tools used for this project. The first section describes the main focus of this research. It also provides a brief description of the project. The next section highlights the resources used in this project. Third section defines the programming language used for this project. It clearly specifies the language and all the tools within that package. Fourth section introduces to the data formats used within this project. The fifth section identifies the environment that was chosen for testing and experiments. It also highlights the pros and cons of each environment. Sixth section provides the main prototype or working model of the project. Seventh section provides the design considerations and reasons for choosing the specific techniques. It has the main block diagram of the project and clearly specifies the flow of data across the model.

The main reason for writing this chapter is to clarify the focus of this research and to present the methodology that is being followed. In-depth explanation of this project is being provided. Physical implementation of the project is being shown.

## 6.1 Research Focus and Overview

The main focus of this research was to develop a system that would take a video input and detect and predict the abnormal behavior in that video surveillance. This project was aimed to monitor traffic, detect abnormal behavior on the railway stations, airports etc. In short, a system that is smart enough to be implemented at any place for abnormal behavior detection. This model can also be used to detect and predict the entry of a person in a restricted area. The focal point of this research was to apply artificial intelligence (AI) techniques to implement the required model.

To implement the research, specific videos were chosen that were clearly targeted towards this application. Furthermore, the choice of videos was based on the light conditions, the amount of information present in that video, the patterns of motion, the speed and timing of the objects within the video.

The main important steps to carry out this research are:

1. To gather the requirements that why this research is being conducted and to make a decision on how this research will be conducted. This task is dependent on the following points:

    i) The choice of environment on which the experiments were to take place

    ii) The choice of model to apply the chosen environment

    iii) The decision on extracted features to be used

    iv) The choice of actions that need to be classified

    v) The decision on surveillance learning technique to detect abnormal behavior

    vi) The choice of data to be provided for training the classifier

    vii) The decision on specifying the areas to be monitored as allowed or not allowed.

2. Implementing the above mentioned points in the form of prototype to perform surveillance for abnormal behavior detection.

3. Training the classifier used in the prototype for testing purposes. Use of this trained data on test videos to further detect and classify the abnormal behavior or anomaly.

To whole research process can be summarized as the Royce's model defined in [**44**]. The process is shown in Figure 6.1.

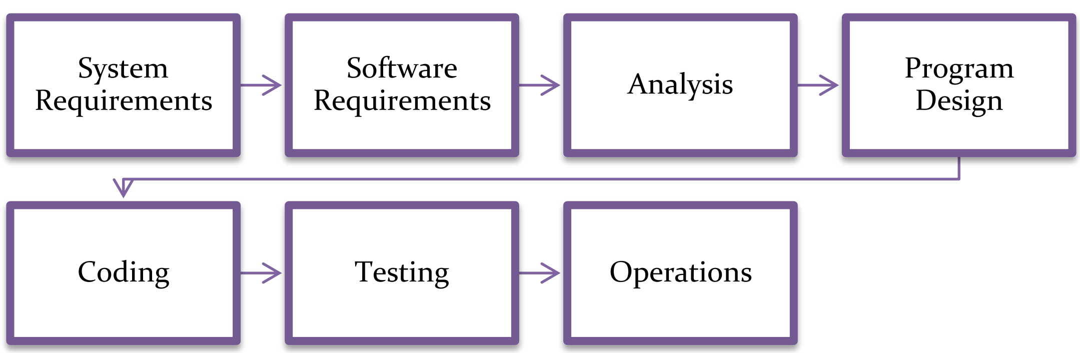



Figure 6.1: Waterfall Method

## 6.2 Provided Resources

Initially a video camera was used to create videos for real time tracking and prediction of abnormal behavior. The video camera recorded the data and stored it on the hard disk. Later on different videos were collected from internet to check the scalability of the project. Furthermore different videos were synthesized using SwishMax to study the motion flow result. The whole video database of project consists of Real time videos from camera and synthesized videos used to compare the results of optical flow output.

The following resources were used for this project:

- **Camera**

  Any webcam or video camera can be used for the videos of this implementation.

- **Software**

  The final implementation is done on MATLAB R2013a (Version 8.1). Visual Studio 2010 is also used for testing purposes.

- **Computer**

  Laptop computer is used with the following specifications:

  - ✓ Processor: Intel(R) Core i3 CPU M350 @ 2.27GHz
  - ✓ Installed Memory(RAM): 2.00 GB
  - ✓ Operating System: 64-bit

## 6.3 Programming Language

The language used in this project is MATLAB. The prototype was initially designed in C# but MATLAB R2012a provided with Computer Vision toolbox that was ideal for this project and later on MATLAB R2013a was used for further enhancements. By using MATLAB, the processing speed also increased. The main toolboxes of MATLAB that are used for this implementation are:

- Image Acquisition Toolbox

- Computer Vision Toolbox
- Neural Network Toolbox

## 6.4 Data Format

Data is present in the form of videos that are real time videos and synthesized videos. The videos are later on converted to images when they are read step by step. After further processing and extraction of features, the features are stored as arrays in *.mat* files.

## 6.5 The Environment to be monitored

The purpose of synthesizing the videos using SwishMax was to observe the results first in a single object environment. All the synthesized videos have white backgrounds with just one object of consideration in a specific type of motion. After testing data and verifying the results, real time videos were tested that contained lots of noise and information. The monitored environment for real time videos was outdoor environment as the aim of this project was to find the optical flow and classify the abnormal behavior on basis of that. The videos were selected carefully so that maximum test cases can be generated from that. Further point that was considered was that there should be at least 4-5 test cases for a single background scene so that the classifier can perform efficiently. Then for the final classification and data monitoring, surveillance videos for different highways at different times was used and the data was extracted to perform regression, curve fitting, pattern recognition and classification.

## 6.6 Prototype Overview

The main prototype of the project is described in Figure 6.2. As shown in the figure, first the video is acquired. The video is then read frame by frame and is divided into image sequences. After that the frames are being fed into a foreground detection system that implements the foregrounding technique using Gaussian Mixture Model. The system outputs a clean foreground mask for each frame that totally removes the non motion objects. This mask is then subject to the optical flow object that has "Lucas-Kanade" as the implementation model and the horizontal and vertical components in complex form so that no information is lost. From this optical flow model, different features are extracted for each frame's horizontal and vertical parts. These features are mean, standard deviation, min, max and variance. These features are fed into Neural Network that initially trains the network by using the features.

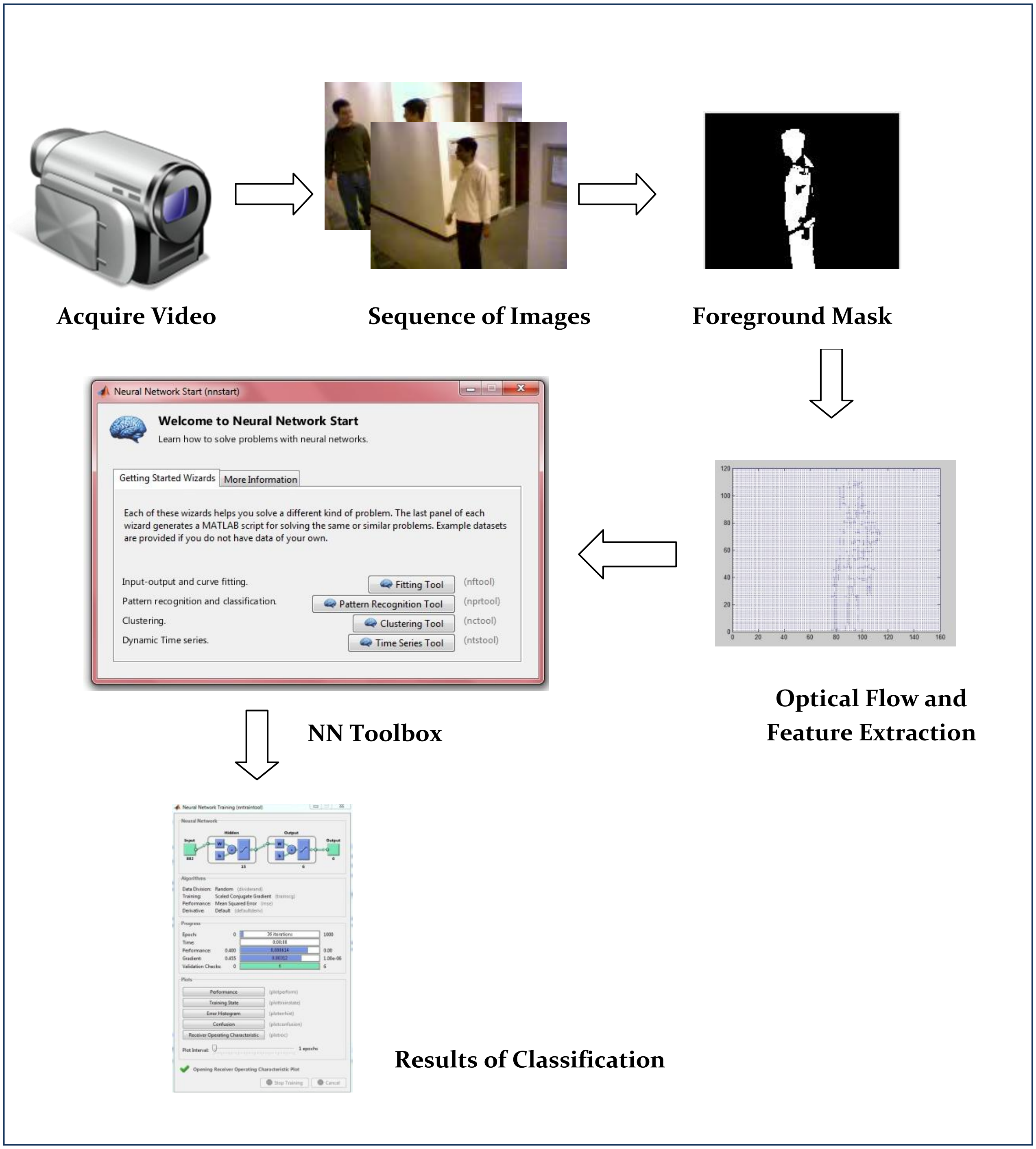



Figure 6.2: Prototype Overview

The features are used as reference for the next incoming test data or video. By combining several test results, a dataset is formed that is being trained for several samples and then is able to be used for classification of data as normal and abnormal. The final result is verified by first looking at the performance of neural network that is obtained by the "perform" command and then graphical results produced are shown in the form of regression, confusion matrix, best validation performance (MSE), training state, error histogram and receiving operating characteristic (ROC).

The next decision that was to be taken was the choice of features to be extracted from the horizontal and vertical components of the optical flow model. The features that have been extracted are mean, standard deviation, min, max and variance. These features are extracted for each frame.

The last decision was the choice of pattern recognition and simulation tool for the features to be identified as belonging to the normal movements and chaotic behaviors. For this purpose, Feed Forward Neural Network was chosen to which data was fed and trained. After training, the incoming video data was being matched with the existing one to classify it as normal or abnormal behavior.

## 6.7 Summary

This chapter has presented with the detailed research focus and implementation of the thesis. The water fall approach used for this project is also being discussed. The resources, tools and programming languages used in the implementation have been discussed. The prototype model has been explained. The data formats and environments have been discussed. In the last section, the design decisions have also been discussed.

# Chapter 7: Implementation

In the previous chapters, all the methods used in the implementation have been discussed. In this chapter, the final architecture or functional model is discussed. The main algorithm is shown in the section below.

## 7.1 Algorithm of Thesis

The algorithm that describes the whole thesis is as followed.

1. Acquire the video.
2. Convert color image to intensity format.
3. Read frames one by one.
4. For every frame:
   i) Apply mean operation to remove the effect of sudden intensity changes.
   ii) Apply foreground mask and Gaussian Mixture Model to the auto balanced image from the previous step.
   iii) Calculate optical flow for each frame using Lucas-Kanade method.
   iv) Extract Horizontal and vertical components.
   v) Extract mean, standard deviation, min, max and variance for the extracted components separately.
   vi) Display the results of optical flow model using quiver and add up to the previous frames result to get the accumulated quiver results.
5. Pass the extracted features to Feed Forward Neural Network after pre-processing the features for NN.
6. Define inputs and targets.
7. Train the network using data from all the results.

8. Simulate for test data to detect normal and abnormal behavior.

9. Generate confusion matrix and plot the results of neural network in the graphical form as regression, confusion matrix, best validation performance (MSE), receiving operating characteristic (ROC).

10. Find error in the generated results in the form of accuracy, overall MSE, sensitivity, specificity, precision, recall, true positive rate and true negative rate.

This algorithm clearly defines each step in the implementation of thesis. The details of the implementation are described below.

## 7.2 Details of Implementation

Videos are stored as 4-D arrays. The first three arrays are for Red channel, Green channel and Blue channel. The fourth array is defined for time. The aim was to develop a model that displays only motion objects and track and classify them. First of all the data to be tracked is fed into the system to be read and is then read frame by frame. Size of each frame is calculated for further processing. The size is an important point of consideration for the optimal results. For increased size and resolution, the effect is on speed and cost of processing the data. Due to the quality of optical flow to detect even the smallest detection, the small video size is the need to develop the model that is fast and efficient.

After that the main aim is to develop the statistical model for the thesis. The steps are discussed one by one in the next sections.

### 7.2.1 Choice of Noise Removal Technique

Due to the fact that optical flow technique can take into account any motion statistics, the main model is first subject to the foreground detector system that mixes up with the Gaussian mixture model to remove the effect of background clutter and to extract the object of interest only. The system extracts only the key information from the video and processes the data. Foreground detection alone cannot be used as it generates noise at some places and is not very effective to take up alone [**28**]. Gaussian Mixture Models are used widely to remove the noisy effects in any image or video processing algorithm. These two techniques are used in combination to produce the results that would be noise free and provide a very refined data to be processed by the optical flow model.

### 7.2.2 Choice of Optical Flow Technique

As the aim was to develop a system that is low in cost, least complex in implementation, fast enough to process and efficient in terms of results for motion only, optical flow technique was the best to be considered for this system. Using optical flow as compared to other methods has the advantage that less data storage is required as compared to other methods. Complexity is reduced as the feature vectors generated by using optical flow are enough to define the motion and objects of interest. Cost of processing is reduced as it requires least bandwidth to transmit only the flow vectors as compared to the whole video that is being monitored. Furthermore, in optical flow the technique that has been used is Lucas-Kanade.

### 7.2.3 Feature Selection

The features that are selected are used for statistical modeling and the decision to choose these features is important as on basis of these features, the classification model will perform. The features, as already described are mean, standard deviation, min, max and variance. These features are extracted for each frame and are stored in an array, the size of which is determined by the number of frames in the sequence. The features are refined using the same computer vision toolbox which has been followed throughout to read the video, foreground detection, optical flow and video display.

### 7.2.4 Choice of Pattern Recognition Tool

The choice of pattern recognition tool was a critical task as on basis of this, the whole classification is to be performed. Neural Network has been selected due to its computational power and the ability to accurately classify the results as based upon the features provided. Neural Network also has the ability to train, validate and test the data and classify it in the form of classes that has been designated.

## 7.3 Choice of Input Videos

The videos that have been chosen for input are all of the sizes stated above in the design decisions (160x120 to 320x240). The main reason for choosing this video size is that fast data processing is the main requirement and optical flow can detect even the smallest of motion so small video size is better that serves the need of this application. As described before, both synthesized and real time videos have been used for test purposes. The snapshots of videos are provided below.

### 7.3.1 Synthesized Videos

Initially some videos were created that contained only single object for test purposes. The videos that have been synthesized using SwishMax for the test purpose of optical flow are:

1. **Car Video:**

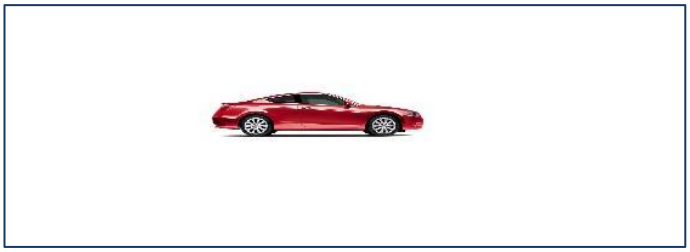


Figure 7.1: Car moving in a straight line

The above Figure shows a car moving in a straight line. This video is of the type "Pan Motion" that is defined in section 2.4.1. The car just moves from left to right.

2. **Ball in the straight line:**

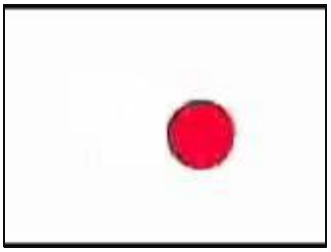


Figure 7.2: Ball moving in a straight direction

The above Figure 7.2 shows a sphere that is moving in a straight line direction. This video is of the same type as above but the difference lies in the shape of object under consideration.

3. **Ball in a circular motion:**

Figure 7.3 shows the snapshots of a video that has a ball moving in circular motion. This video is an example of rotation motion. It has a bell that moves clockwise.

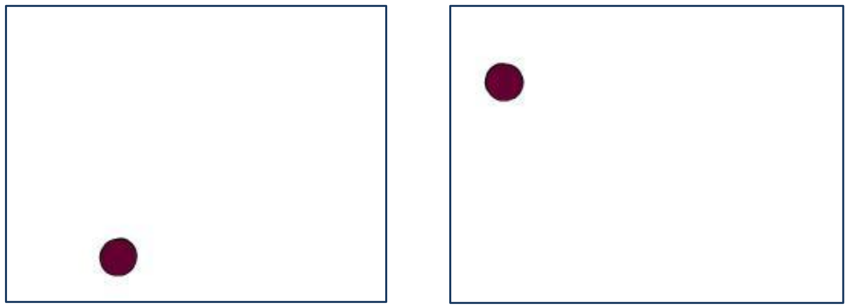


Figure 7.3: Ball moving in a circular motion

### 7.3.2 Real Life Videos

This section will provide the snapshots of videos from real life.

1. **Cars moving in a single direction**

Figure 7.4: Cars moving in a single direction

Figure 7.4 shows two sequences from the video in which cars are moving in the downward direction. There are multiple cars of mixed colors and they are all moving in one direction one after other.

**2. Two men entering into a room**

Figure 7.5: Two men entering into a room

The figure above shows two men entering in a room. They enter the room one by one, shake hands and then leave the room.

**3. Cars accident on road from backside**

Figure 7.6: Car bumped from back side

The above figure shows a sequence in which a car stops on the signal and after a few seconds, another car comes from backside and hit the stopped car.

**4. Cars accident, head on collision**

Figure 7.7: Two cars in head on collision

The above figure shows the snapshots of a video in which two cars that are coming from opposite sides, collides from front and both slide on the roadside.

5. **Road Junction Surveillance**

Figure 7.8 shows some snapshots of a multi-junction signal where multiple roads meet the same point. This video is has all types of data as per related to traffic. There are parts when the traffic is moving normally and there are parts when there is an accident on the road or some collision between the cars. The probability of accidents is highly decreased by road junctions and signal lights, but there are few cases, as shown in video where accidents happen due to negligence, hurry or just luck.

**(a)**

**(b)**

**(c)**

**(d)**

Figure 7.8: (a), (b) Normal Road Traffic Snapshot, (c), (d) Accidents on the Road

## 7.4 Videos Dataset

Apart from the videos shown above, there were many videos that were tested. A complete dataset is provided in the CD accompanying the dissertation. The dataset consists of a total of 55 videos. Some of the videos were modified according to the need of application; the dataset however has the original videos. The dataset consists of videos from MATLAB vision library, videos downloaded from YouTube and the videos that were made using SwishMax.

## 7.5 Summary

This chapter described the whole implementation model in terms of algorithm and choice for implementation. The first section described the main implementation along with the algorithm for the thesis. The next section focused on the details and reasons of choosing each technique implemented. All the parts of implementation are discussed separately to provide brief insight of the working model. The last section dealt with the input videos that have been used for the algorithm. All types of videos used, have been discussed in detail.

# Chapter 8: Results and Analysis

In this chapter, all the results that have obtained from testing and implementation are provided. The chapter starts with the results of optical flow implementation without using foreground extraction. The next section shows the results of foreground extraction with GMM. It then proceeds towards the explanation of results produced by applying the foreground extraction results to optical flow model. The last section shows the results produced by pattern recognition and classification using neural network.

## 8.1 Original Optical Flow Results

This section shows the original results of optical flow implementation. Results for the videos that have been shown in section 7.3 are being shown here.

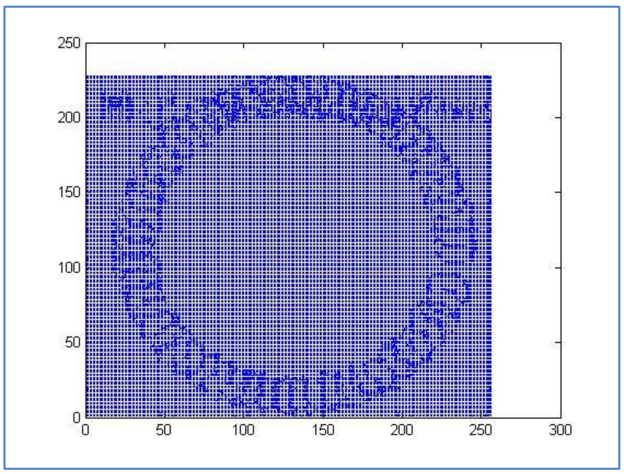



Figure 8.1: Quiver results of ball moving in circular motion

The figure above shows the result of applying optical flow model directly to the video that contains a ball moving in a circular motion. As evident from the result, on the top of the figure, noise can be seen. This noise is the result of text appearing on top of the video.

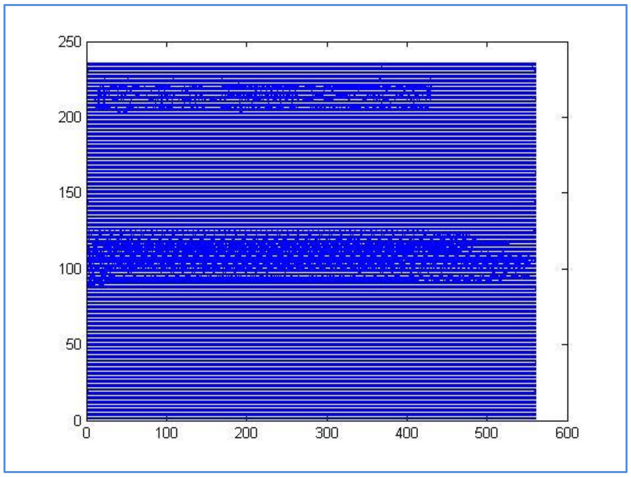



Figure 8.2: Quiver results of car moving in straight line

Figure shows the result of applying the model to the video that contains a car moving in a straight line. Here again, on the top of the figure, noise can be seen. This noise too is the result of text appearing on top of the video.

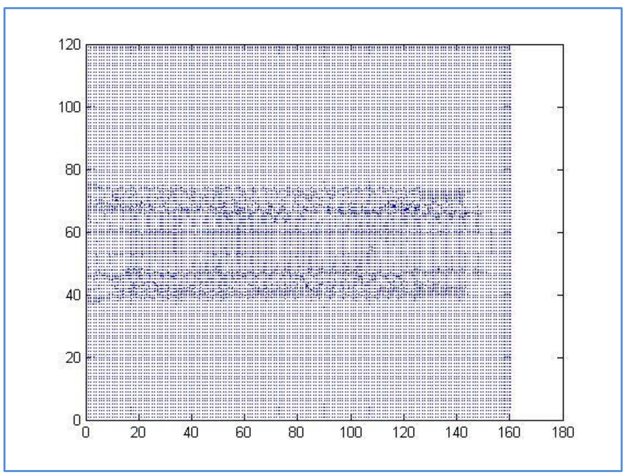



Figure 8.3: Quiver results of ball moving in straight line

The figure above shows the results of a ball moving in a straight line. As seen from the image, the optical flow vectors are more scattered and contain a lot of details. Just by looking at that, it can be deduced that the vectors are needed to be refined.

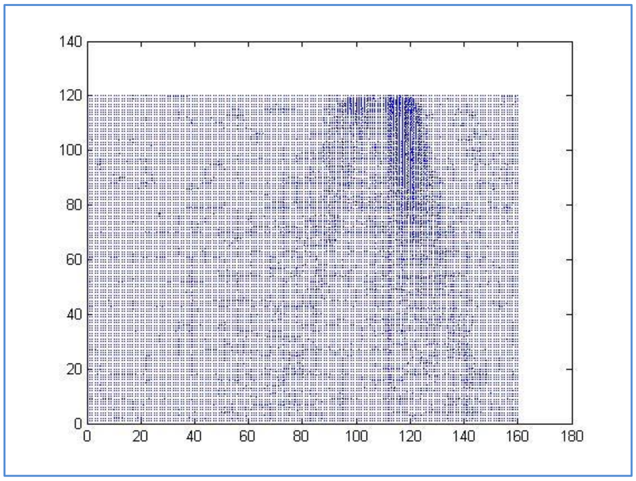



Figure 8.4: Quiver Results of cars moving in a single direction

Figure 8.4 shows the result of cars moving in single direction. The result shows that vectors of sideways and trees etc have also been computed. The result due to non-motion is actually noise over here and is not needed. Again, the flow vectors are scattered here too.

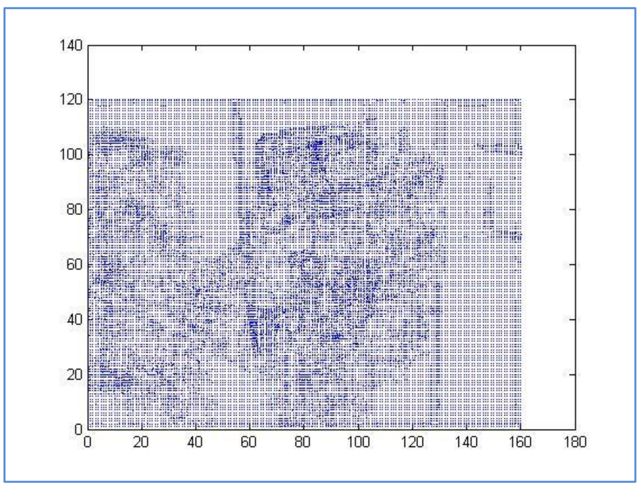



Figure 8.5: Quiver Results of Two men entering into a room

Above figure shows the results of two men entering in a room. Here it is seen that the details of walls and non static objects have also been taken into account.

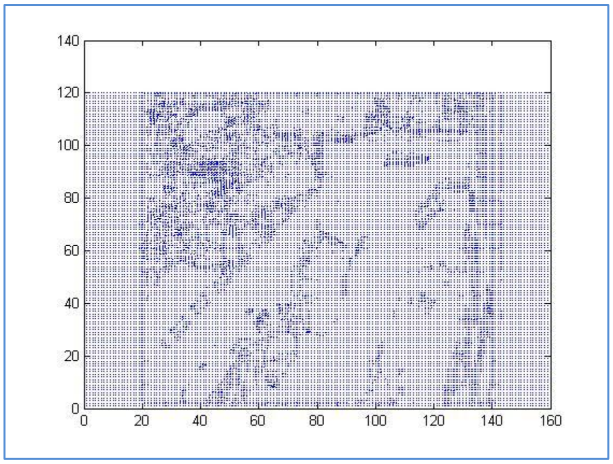



Figure 8.6: Quiver results of two cars in head on collision

Figure 8.6 shows the result of two cars colliding from front. The quiver plot shows the sideways and static data as well that is not required.

All the figures that are shown above have the results for optical flow alone. These are the results that have been computed before using foreground detection. As shown in the figures, it is clearly indicating the noise that is also been incorporated in the output sequence. The main aim was to detect only the objects that are in motion and process them only. To cater for this, foreground detection with GMM is used. This implementation is described in the next section.

## 8.2 Foreground Detection Results

This section shows the original results of foreground detection for the videos that have been shown in section 7.3.

Figure 8.7: Results of ball moving in straight line

The figure above shows the result of applying foreground with GMM (FGMM). It can be seen that the all the extra data has been removed and no noise is present in the output.

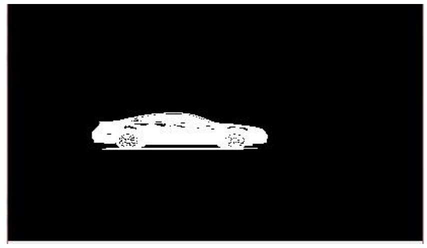


Figure 8.8: Results of car moving in straight line

Figure 8.8 shows the same results for the car video. Here the object of interest, car, is extracted only with no shape deformation.

Figure 8.9: Results of two cars head on collision

The above figure shows only the two cars colliding, eradicating the background noise.

Figure 8.10: Results of ball moving in a circle

Figure above shows the result of FGMM on the ball moving in circular motion. Here the object of interest is extracted with minimum noise.

Figure 8.11: Results of two men entering into room

Figure 8.11 shows the result of two men entering into room. In the snapshot, there is only one person and that is the object of interest. It can be seen that only the person is extracted with no background noise.

Figure 8.12: Results of cars moving in one direction

The figures above show the result of cars moving in one direction. Only the cars are extracted with no background noise.

## 8.3 Results after Application of FGMM to Optical Flow Model

This section shows the final results of quiver after FGMM and optical flow. All the figures below are the next level implementation of results explained in previous section.

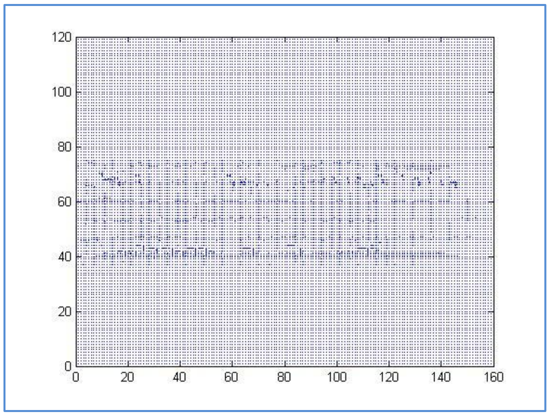



Figure 8.13: Results of ball moving in straight line

For the video that has a ball moving in a straight line, the result of applying FGMM to optical flow model for this has been shown above. As seen, the top text that was appearing as noise has been eliminated. The vectors are also reduced in size and only the motion vectors of interest have been generated. The size of data is reduced considerably and only the useful information is extracted.

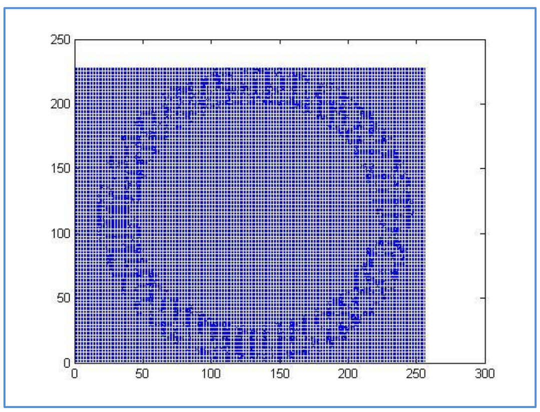



Figure 8.14: Results of ball moving in a circle

The above figure shows the result of applying FGMM to optical flow model for a ball moving in a circular motion. For this too, the top text that was appearing as noise has been eliminated. Only the trajectory of motion and shape is being extracted with the minimal number of motion vectors.

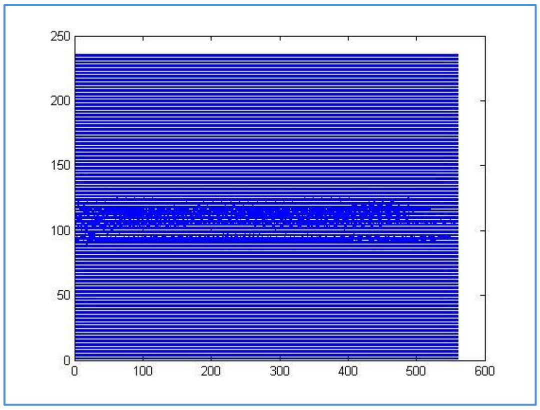



Figure 8.15: Results of car moving in straight line

Figure 8.15 shows the results of car moving in straight line. Here the text on top has been removed and the trajectory of data is being shown. Again, minimal number of useful vectors has been generated.

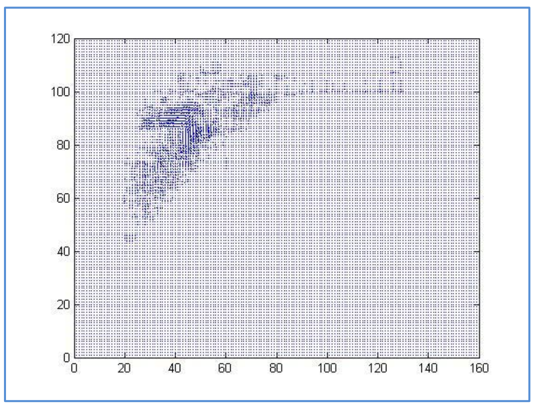



Figure 8.16: Results of two cars head on collision

The figure above shows the results of two cars colliding from front. Here, only the cars optical flow vectors have been extracted and not the surroundings. This presents the minimal requirement of feature extraction and data extraction is a lot easier. The speed of processing is also increased.

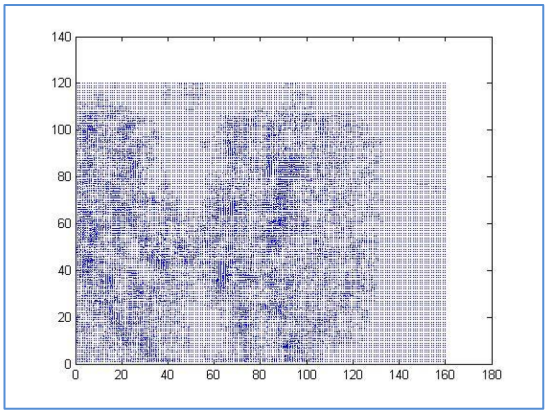



Figure 8.17: Results of two men entering into room

Figure 8.17 shows two men entering in a room. From the figure, it can be analyzed that only the men have been extracted that are objects of interest and no noise is present in the data.

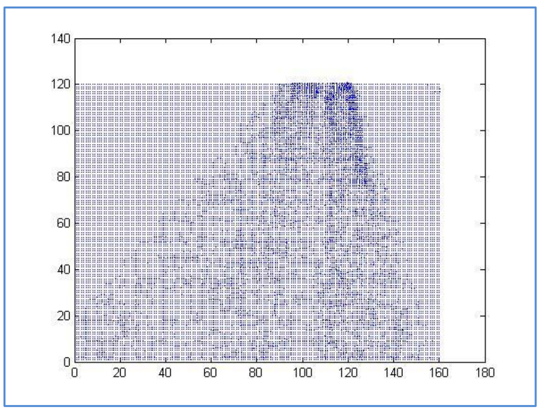



Figure 8.18: Results of cars moving in one direction

The above figure shows the results of cars moving in a single direction. The trajectory, as can be seen, is confined in a single direction and only the cars motion vectors have been extracted.

The figures above present the final results as computed from the algorithm. These are the results that are presented to the classifier and are recognized. As the results show, it can be clearly seen that there is no noise in the models as was present in the results of section 8.1.

## 8.4 Results of Neural Network

The neural network takes the features and then classifies them on the basis of description provided. The type of neural network that has been used is *feed forward network*. In this network, the data moves only in one direction, i.e., from input to output. The network has input layer, hidden layers and output layers. The network has been tested using different numbers of hidden layers and optimized results has been found after much testing on manipulating the input data, the hidden layers, the transfer functions and epochs. The videos shown above and a few more videos data has been presented to the neural network. For the final test purpose, the video shown in section 7.3 that has the surveillance record of a multi-junction road has been used. This video is being observed for both normal and abnormal test cases. The normal test cases are defined as the cases in which the traffic is moving normally with no accidents and no violence of rules. The abnormal behavior is the case when there is a sudden chaos on the road or an accident occurs. Both videos have been treated separately.

The data is then combined to create a large data set that is presented to the neural network for further testing. This data is then arranged in two classes, one being normal data and other being abnormal. This arrangement is done using *targets* in neural network. The neural network then classifies the data and performs training of the data. After training, the data is validated and then tested for new inputs that were not presented to the network before. After trying different data sets and network functions, graphical outputs are generated in the form of confusion matrix, performance, validation results, error histograms and receiver operating characteristics. Apart from graphical outputs, the final output is also calculated in term of a parameter *performance* that is actually the MSE.

As the final real time data, all the tests are performed on the multi-junction road surveillance data as explained before. These final results that have been obtained with all the optimized parameters are presented in the next sub sections.

### 8.4.1 Main Neural Network Model

The main model that has been created is shown in Figure 8.19 below. As the above model shows, the network has 5 input feature vectors. The whole data set is a **5 x n** matrix where **n** is the number of samples. The output is a **1xn** matrix, n samples of 1 element which is one class (normal or abnormal). The results have been calculated by changing the number of hidden layers and by changing the function for neural network to *trainscg*.

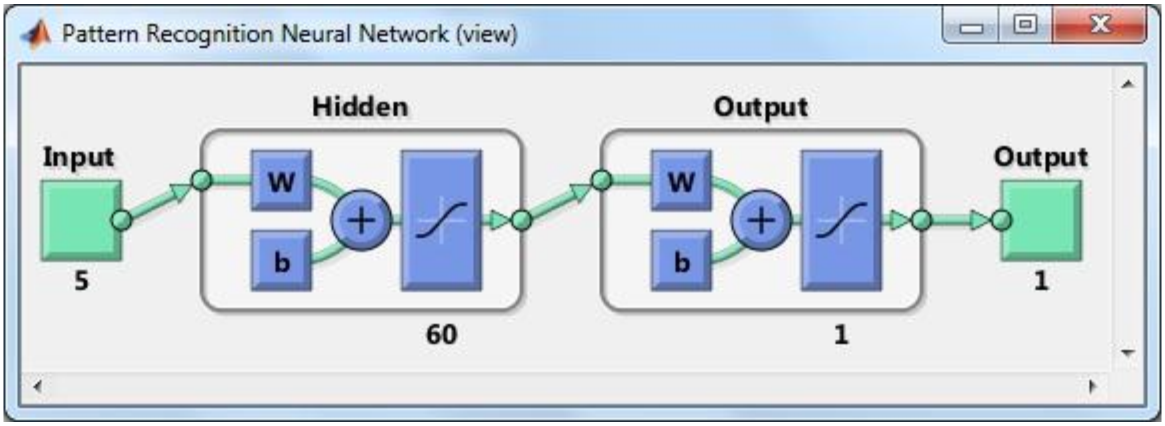



Figure 8.19: Main Neural Network Model for Application

For this application, very good performance for the network with a value of 3.5129e-02 has been obtained. The overall error in the system is found to be 2.5%.

### 8.4.2 Performance Plot

Performance plot is basically the result of neural network in the form of graph that shows the performance of the net over the epochs. The performance plot is shown in Figure 8.20.

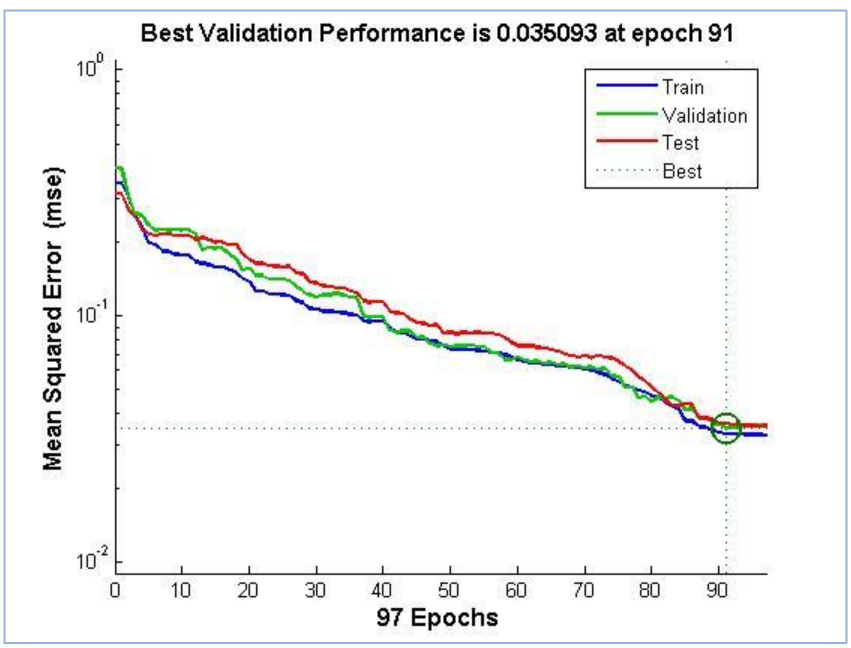



Figure 8.20: Performance Plot of the Neural Network

As can be seen from the figure, the performance of the neural net for training, testing and validation is constantly increasing with growing epochs. The MSE is decreasing which indicates that the network is performing well. As seen from the above, the best result is found at epoch number 91 which produces the performance result as **3.5 e-02**, which is a fairly good result.

### 8.4.3 Receiver Operating Characteristics

An ROC is basically a test to measure the accuracy of results. The receiver operating characteristics should ideally be directed towards the left and top of any graph. The main concept of ROC is defined in [**45**].

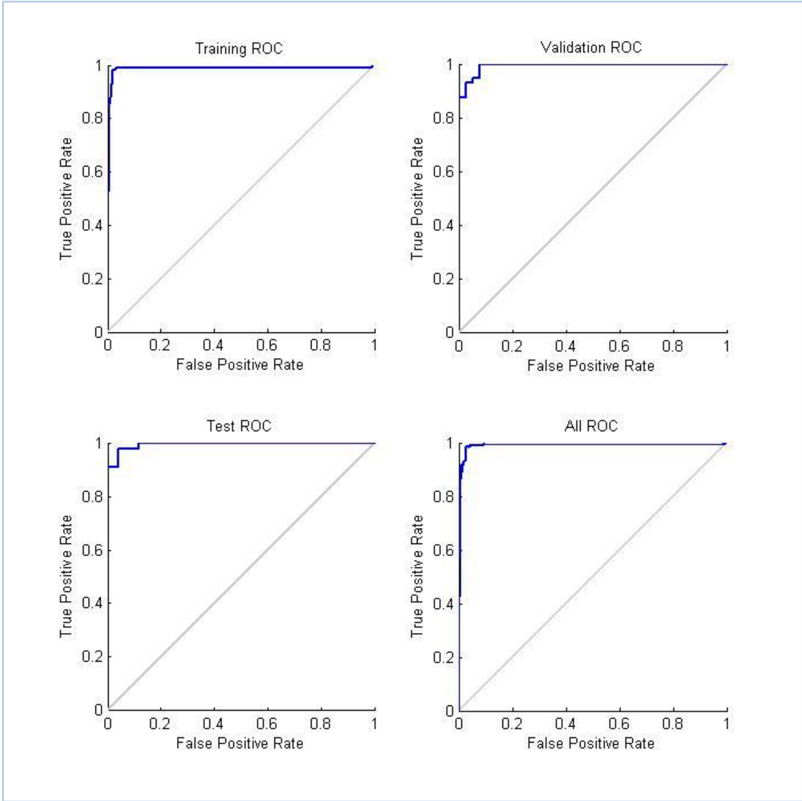



Figure 8.21: ROC for Neural Network

As can be seen from the figure above, the curve is directed to the left and top and is showing the expected results for classification.

### 8.4.4 Confusion Matrix

Confusion matrix is a test to verify the performance of the neural net or to see that how well the data is fit to the network. It shows the percentage of classifications that are correct and incorrect.

The classifications that are correct are shown in green boxes and the ones that are misclassified or wrongly classified are shown in red boxes. The percentage of classifications in the red boxes should be small if the network has learned to classify properly.


Figure 8.22: Confusion Matrix for Training, Validation and Test Data

The matrix above shows the results of training, validation and testing data. The main matrix is on the bottom right which shows the data of all the classifications. As seen from the matrix, the results are:

- 311 samples from class 0 were correctly identified (Results in green)
- 9 samples from class 0 were misclassified(Results in red)
- 313 samples from class 1 were correctly identified(Results in green)
- 7 samples from class 1 were misclassified(Results in red)

From the results of the bottom right matrix, it can be seen that overall accuracy of the system is 97.5% with an error percentage of 2.5%. These percentages show that majority of data has been correctly classified. With a few errors in the data, the classes are properly classified and further accuracy has been calculated using different parameters that are explained in the next section.

### 8.4.5 Performance Parameters

The parameters calculated using different formulas and from MATLAB code output are explained below. Details of all these parameters were discussed in chapter 5.

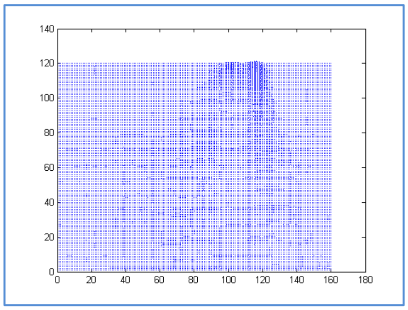
Table 8.1: Performance Parameters of the Model

| Parameter | Result |
|---|---|
| **Sensitivity (Recall)** | 97.19% |
| **Specificity** | 97.81% |
| **False Positive Rate** | 2.19% |
| **False Negative Rate** | 2.81% |
| **% Wrong Classified** | 2.5% |
| **Precision** | 97.8% |
| **Accuracy** | 97.5% |
| **F-Measure** | 0.97 |

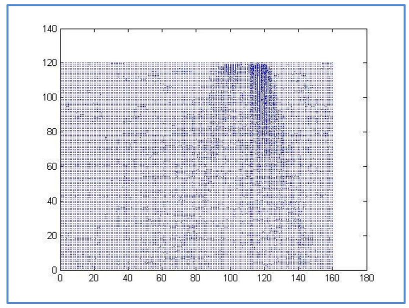
Above results show the accuracy of model. For overall results, we have an accuracy of 97.5% that is the MSE. A percentage error of 2.5 % is obtained for the whole model. The range of score is 0-1, where 0 is for worst performance and when we go towards 1, we have the best performance. The F-Score is calculated as 0.97 which shows very good results.

## 8.5 Comparison with Previous Techniques

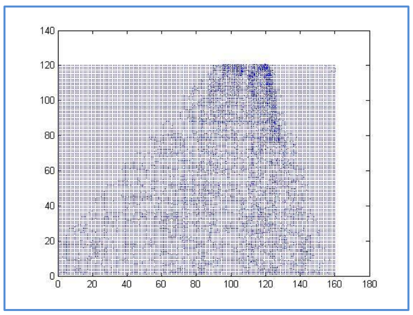
This section provides a comparison of the traditional Horn-Schunck Method [**13**], Lucas-Kanade Method [**14**] and our proposed technique. The first comparison is shown in graphical form. The comparisons are shown for a single video sequence for the sake of scarcity.




(a) Original Lucas-Kanade Results




(b) Original Horn-Schunck Results




(c)Results of Proposed Technique

Figure 8.23: Compassion of Lucas-Kanade Method and Proposed Technique

The figures clearly show that the proposed technique provides noiseless results that are very smooth and defined as shown in Figure 8.23(c). The original Lucas-Kanade method and Horn-Schunck method introduced some noise or unwanted results.

After providing the neural network with the features extracted from the techniques above it has been found that the original Lucas-Kanade method showed an accuracy of 86.7% and an overall error percentage was found to be 13.3%. The original Horn-Schunck method showed an accuracy of 73.8% and an overall error percentage was found to be 26.2%.The proposed technique presented an overall performance of 97.5% with an error percentage of just 2.5%.

The MSE for proposed method is calculated is 3.5 e-02 whereas it was found to be 1.21 for the Lucas-Kanade technique and 1.52 for Horn-Schunck technique.

Confusion matrices for the original Lucas-Kanade method and Horn-Schunck method are shown in Figure 8.24.



(a) (b)

Figure 8.24: Confusion Matrices for Original Lucas-Kanade and Horn-Schunck Method

The above matrices show the poor performance of both Lucas-Kanade and Horn-Schunck methods as compared to the proposed method. Table 8.2 shows the comparison of performance parameters of the three methods; Lucas-Kanade, Horn-Schunck and proposed method. It is clearly shown that the proposed model performs best.

Table 8.2: Comparison of Performance Parameters of Proposed Model with Original LK and HS Methods

| Parameter | Lucas-Kanade | Horn-Schunck | Proposed Method |
|---|---|---|---|
| **Sensitivity (Recall)** | 86.6% | 67.2% | 97.19% |
| **Specificity** | 86.9% | 80.3% | 97.81% |
| **False Positive Rate** | 13.1% | 19.7% | 2.19% |
| **False Negative Rate** | 13.4% | 32.8% | 2.81% |
| **% Wrong Classified** | 13.3% | 26.2% | 2.5% |
| **Precision** | 86.8% | 77.3% | 97.8% |
| **Accuracy** | 86.7% | 73.8% | 97.5% |
| **F-Measure** | 0.87 | 0.72 | 0.97 |

The above table clearly shows that the performance of the original LK and HS methods has been very poor as compared to the proposed method. It can also be seen that Horn-Schunck method performed very poor as compared to Lucas-Kanade method. Keeping these results in view, modification has been made to the Lucas-Kanade method. Features extracted from the two methods were not as good as the proposed method and hence generation of low F-Measure was observed.

From all above results that the proposed model has produced results near to ideal and have performed the best as compared to the other two methods.

## 8.6 Summary

In this chapter, the results of whole thesis have been shown. The first section showed the quiver results of optical flow and noise free optical flow has been shown. In the next section, neural networks results has been shown and specified. The results have been shown in graphical as well as parametric form.  The last section was based on comparison between the original Lucas-Kanade method, Horn-Schunck method and the proposed technique.

# Chapter 9: Conclusions and Future Work

## 9.1 Conclusions

This thesis has been aimed to automatically find the abnormal behavior within any surveillance data. The aim was to develop a system that correctly finds the optical flow for real time environment and classify the results using a pattern recognition technique as being normal or chaotic. This has been achieved by testing different algorithms for optical flow and parameters. Using optical flow as the ground technique and neural network as a classification technique, a model is developed to correctly estimate the abnormal behavior. The uniqueness of this thesis lies in the fact that here foreground detection system is being implemented before the optical flow system and hence a noise free model is obtained from the optical flow results. Both, real time and synthesized videos has been tested and lowest error rates and correct classification has been obtained.

In the following sub sections, first the initial goals and objectives have been reviewed. After that comes the noise removal technique for non-motion elimination. The next section describes the optical flow technique. Last section defines the pattern recognition and classification technique.

### 9.1.1 Initial Goals and Objectives

Before presenting the final conclusion, overview of the implemented techniques is given that provides with the details of the results and achievements. The main aim was to develop a model for video surveillance that automatically detects the abnormal or chaotic behavior in the real time data.

- Eliminating Non-Motion Effect:

    1. Only the objects in motion should be detected.
    2. There should not be any outliers or distortion due the surrounding objects.
    3. A proper foreground model should be extracted before that contains only the objects in motion.
    4. Technique for the foreground extraction should be devised.

- Optical Flow Technique Implementation:

    1. Selection of the technique appropriate for the implementation.
    2. Comparing the two main techniques i.e. Lucas-Kanade and Horn–Schunck.
    3. Finding optimal parameters out of these two for the implementation.
    4. Generating the results in the form of motion vectors that contain the trail of previous frame results as well. In other words, trajectory is generated for the flow of data.
    5. Performing further enhancement techniques for smoothing optical flow and selecting the best features from the results to pass it on to the classifier.

- Feature Selection:

    1. Selecting the best features that describe the objects in motion correctly.
    2. Pre-processing the data to filter out the best features.
    3. Saving the features as input data set and first manually labeling the features as belonging to normal or abnormal class for training.

- Pattern Recognition Tool Selection:

    1. Selecting the pattern recognition technique best suited for the detection of normal and abnormal behavior using the feature set generated from the optical flow model.
    2. Passing the data to the classifier
    3. Post-process the output for the performance estimation results.

### 9.1.2 Elimination of Non-Motion Effect

The aim was to develop a model that should first separate the moving and non-moving objects. For this purpose, foreground extraction has been. The issue faced was that foreground detection alone could not remove the noise entirely. So for this purpose, Gaussian Mixture Model has been used as base line method for foreground extraction, which correctly pulls out the structure of the object in motion and removes all other objects that are considered as noise for this application. Using this technique, the effect of non-motion objects has been eliminated.

### 9.1.3 Optical Flow Technique Implementation

As the aim was to develop a system that is low in cost, least complex in implementation, fast enough to process and efficient in terms of results for motion only, optical flow technique was the best to be considered for this system. Using optical flow as compared to other methods has the advantage that less data storage is required, complexity is reduced as the feature vectors generated by using optical flow are enough to define the motion and objects of interest, cost of processing is reduced as it requires least bandwidth to transmit only the flow vectors as compared to the whole video that is being monitored. Furthermore, in optical flow the technique that has been used is Lucas-Kanade (LK). For this application, LK performed better as it divides the video frame in small portions and then solves it as compared to Horn-Schunck that assumes that optical flow is smooth over the entire image. Optical flow is computed by using the model generated from the foreground extraction and the motion vectors are generated only for the objects in motion.

### 9.1.4 Feature Selection

Feature selection was the most crucial step as depending on these features, results were to be produced by the classifier and pattern recognition tool. The features that were used are mean, standard deviation, min, max and variance. These features are extracted for each frame and are stored in an array, the size of which is determined by the number of frames in the sequence. The features are refined using the pre-processing techniques for neural networks that include normalization. Features for both normal and abnormal data are stored separately which are later on combined after all the steps of pre-processing.

### 9.1.5 Pattern Recognition Technique Selection

Choice of pattern recognition tool was another critical task as on basis of this, the whole classification was to be performed. Feed Forward Neural Network has been selected due to its computational power and the ability to accurately classify the results as based upon the features provided. Neural Network also has the ability to train, validate and test the data and classify it on basis of classes that has been designated. The pre-processed features are passed on to the neural network and the data is presented in the form of inputs and targets. Targets are the desired outputs for each input.

The data has been trained to adjust the weights of the network for accurate results until the desired or acceptable Mean Squared Error (MSE) is obtained. Different parameters including changing of data set were altered during this phase. After that, validation was performed that is used for overfitting minimization. Validation determines the criteria to stop training. Finally the test data was presented to the network that checked whether the network is performing well or not and from here, the classification results in the form of normal and abnormal data has been obtained. The results obtained are confusion matrix, best validation performance (in the form of MSE), receiving operating characteristic (ROC), Accuracy, Precision and Recall, F-score, Percentage Error and Finally the overall performance of the network.

## 9.2 Obtained Results

The results that were obtained in the end using neural network has been estimated as 97.5% accurate with just 2.5% error percentage. The overall performance is calculated by MSE and is found to 3.5e-02. The F-measure is calculated as 0.97.

## 9.3 Future Work

This thesis can be extended for high resolution videos as well. Furthermore, the features can be changed using any feature selection technique and different tests can be performed by using other pattern recognition techniques for the comparison of performance. The optical flow model can be used in conjunction with other techniques of noise removal that may generate better results.

Further work for the improvement of algorithm can be done on the videos that have been used for this implementation. Using the same dataset as of this will be helpful for quantitative analysis with the results presented.

# References


[1] L. G. S. a. G. C. Stockman, *Computer Vision*. Prentice Hall, 2001.

[2] T. Morris, *Computer Vision and Image Processing*. Palgrave Macmillan, 2003.

[3] P. Corke, *Robotics, Vision & Control Fundamental Algorithms in MATLAB*, B. Siciliano and O. Khatib, Eds. Springer, 2011, vol. 73.

[4] C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications*, 1st ed. Wiley-VCH, December 26, 2007.

[5] J. Gilvarry, "Extraction of Motion Vectors from an MPEG Stream," Dublin City University, Technical Report, 1999.

[6] R. Malhotra, "Detection and Segmentation of moving objects in video using optical flow estimation," MSc Thesis, Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, 2008.

[7] X. Chen, X. Li, H. Wu, and T. Qiu, "Real-time object tracking via CamShift-based robust framework," in *International Conference on Information Science and Technology (ICIST)*, 2012, pp. 527-530.

[8] W. Hu, et al., "Active Contour-Based Visual Tracking by Integrating Colors, Shapes, and Motions," in *IEEE Transactions on Image Processing*, 2013, pp. 1778-1792.

[9] S. Torkan and A. Behrad, "A new contour based tracking algorithm using improved greedy snake," in *18th Iranian Conference on Electrical Engineering (ICEE)*, Isfahan, Iran, 2010, pp. 150-155.

[10] h. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, San Francisco, 2010, pp. 2054-2060.

[11] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona , 2011, pp. 1419-1426.

[12] J. J. Gibson, *The Perception of the Visual World. Houghton Mifflin.* 1950.

[13] B. K. P. Horn and B. G. Schunck, "Determining optical flow," in *Artificial Intelligence*, 1981, p. 185–203.

[14] B. D. L. a. T. Kanade, " An iterative image registration technique with an application to stereo vision.," in *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121--130.

[15] S. Roth and M. J. Black, "On the spatial statistics of optical flow," in *Tenth IEEE International Conference on Computer Vision*, vol. 1, 2005, pp. 42-49.

[16] M. Liu, C. Wu, and Y. Zhang, "Motion vehicle tracking based on multi-resolution optical flow and multi-scale Harris corner detection," in *IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 2032-2036.

[17] X. Zhu, J. Liu, J. Wang, Y. Fang, and H. Lu, "Anomaly detection in crowded scene via appearance and dynamics joint modeling," in *19th IEEE International Conference on Image Processing (ICIP)*, Orlando, FL, 2012, pp. 2705-2708.

[18] M. S. Kemouche and N. Aouf, "A Gaussian mixture based optical flow modeling for object detection," in *3rd International Conference on Crime Detection and Prevention*, London, 2009, pp. 1-6.

[19] W. Li, X. Wu, K. Matsumoto, and H.-A. Zhao, "Foreground detection based on optical flow and background subtract," in *International Conference on Communications, Circuits and Systems* , Chengdu, 2010, pp. 359-362.

[20] S. Elie, "An overview of Pattern Recognition," University of Burgundy, Antonine University Article, April, 2013.

[21] U. Jänen, C. Paul, M. Wittke, and J. Hähner, "Multi-object tracking using feed-forward neural networks," in *International Conference of Soft Computing and Pattern Recognition*, Paris, 2010, pp. 176-181.

[22] T. Emoto, M. Akutagawa, U. R. Abeyratne, H. Nagashino, and Y. Kinouchi, "Tracking the states of a nonlinear system in the weight-space of a feed-forward neural network," in *IEEE International Joint Conference on Neural Networks*, 2005, pp. 96-101.

[23] W. Diffie and S. Landau, "Internet Eavesdropping: A Brave New World of Wiretapping," *Scientific American*, Aug. 2008.

[24] (2009, Mar.) Tracking a suspect by mobile phone.

[25] M. Irani, B. Rousso, and S. Peleg, "Recovery of Ego-Motion Using Image Stabilization," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, p. 21–23.

[26] S. Liu, "Object Trajectory Estimation Using Optical Flow," 2009.

[27] M. Piccardi, "Background subtraction techniques: a review," in *IEEE International Conference on*

*Systems, Man and Cybernetics*, October 2004, p. 3099–3104.

[28] T. Bouwmans, F. E. Baf, and B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey," in *Recent Patents on Computer Science* , 2008, pp. 219-237.

[29] Z. H. Khan, I. Y.-H. Gu, and A. G. Backhouse, "A Robust Particle Filter-Based Method for Tracking Single Visual Object Through Complex Scenes Using Dynamical Object Shape and Appearance Similarity," *Journal of Signal Processing Systems*, vol. 65 , no. 1, pp. 63-79, Oct. 2011.

[30] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley, 2009.

[31] P. Aimonen. (2011, Nov.) Wikipedia. [Online]. http://en.wikipedia.org/wiki/File:Basic_concept_of_Kalman_filtering.svg

[32] O. MARQUES, *PRACTICAL IMAGE AND VIDEO PROCESSING USNG MATLAB*. John Wiley & Sons, Inc, 2011.

[33] K. R. T. Aires, "Optical flow using color information: preliminary results," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 1607-1611.

[34] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," in , New York, USA., 1995.

[35] R. Milewski and V. Govindaraju, "Binarization and cleanup of handwritten text from carbon copy medical form images," in *Pattern Recognition*, 2008, p. 1308–1315.

[36] S. O. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, November 28, 2008.

[37] D. J. Fleet and Y. Weiss, "Optical Flow Estimation," in *Mathematical Models in Computer Vision: The Handbook*. Springer, 2005, ch. 15, pp. 239-258.

[38] T. Huang, J. Qiu, and T. Ikenaga, "A Foreground Extraction Algorithm Based on Adaptively Adjusted Gaussian Mixture Models," in *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference*, Seoul, 25-27 Aug. 2009.

[39] K. Toyama, J. Krumm, B. Brummit, and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," in *ICCV*, Corfu, Greece, 1999.

[40] C. Stauffer and W. E. L. Grimson, " Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, p. 246–25.

[41] T. Bouwmans, F. E. Baf, and B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey," in *Recent Patents on Computer Science*, 2008, pp. 219-237.

[42] Bain, "Mind and Body: The Theories of Their Relation.," 1873.

[43] James, "The Principles of Psychology. ," 1890.

[44] D. W. W. Royce, "Managin the Development of Large Software Systems," in *Proceedings of IEEE WESCON 26 (August)*, 1970, pp. 1-9.

[45] K. Woods and K. W. Boyer, "Generating ROC Curves for Artificial Neural Networks," in *IEEE TRANSACTIONS ON MEDICAL IMAGING*, vol. 16, NO. 3, 1997.