Object Detection and Video Surveillance System

Submitted By

Hetali Tank

13MCEN35

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Object Detection and Video Surveillance System

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Hetali Tank

(13MCEN35)

Guided By

Dr. Sanjay Garg

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2015

Certificate

This is to certify that the major project entitled **"Object Detection and Video Surveillance System"** submitted by **Hetali Tank (Roll No: 13MCEN35)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Networking Technology) of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Gaurang Raval
Associate Professor,
Coordinator M.Tech - CSE(NT) Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Guide, Professor and Head,
CSE Department
Institute of Technology,
Nirma University, Ahmedabad

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

Statement of Originality

I, **Hetali Tank**, Roll. No. **13MCEN35**, give undertaking that the Major Project entitled "**Object Detection and Video Surveillance System**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Networking Technology)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student
Date:
Place:

Endorsed by
Dr. Sanjay Garg
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Sanjay Garg**, Guide and Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance, kind support, providing basic infrastructure and healthy research environment and continual encouragement throughout this work. The appreciation and continual support they has imparted has been a great motivation to me in reaching a higher goal. Their guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Mr. Chirag Patel** & **Prof. Vijay Ukani**, for his kind support and guidance throughout this work.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Hetali Tank
13MCEN35

Abstract

Now a days, Video Surveillance systems are quickly becoming a key component of the security system. Making an intelligent video surveillance system, which can able to detect and track multiple objects and also deals with illumination problem and whether conditions(fog, rain, snow etc) is quite challenging task. There are many applications and areas in which Video surveillance is necessary. Video Surveillance in dynamic scenes is active research area in computer vision and there are many applications in which video surveillance is necessary for security, monitoring purpose. From the various application proposed system is designed for real time automatic vehicle detection and vehicle counting from stationary background. Nowadays, we see that the traffic is increases day by day on roads, highways due to increase in number of vehicles. Vehicle detection, classification and counting is very important application by which highway monitoring, traffic planning, analysis of the traffic flow, etc can be easily done. In this thesis, computer vision based moving vehicle detection and counting is presented. By the background subtraction method moving vehicle is detected in ongoing video. And further analysis is done to classify the moving vehicle as car, bike, truck etc. The system is implemented using MATLAB and experimental results are demonstrated on the dataset as well as real time video taken from the static camera. The system is also capable for counting the number of vehicles.

Key Terms: Video Surveillance, Object Detection, Object Classification, Background Subtraction.

Abbreviations

BGS	Background Subtraction.
MOG	Mixture Of Gaussian.
AMS	Approximated Median Subtraction.
FD	Frame Difference.
PCA	Principal Component Analysis.
FCH	Fuzzy Color Histograms.
GMM	Gaussian Mixture Model.
SVM	Support Vector Machine.
ANN	Artificial Neural Network.
BBN	Bayesian belief network.
SIFT	Scale Invariant Feature Transform.
HOG	Histograms of Oriented Gradient.
DOG	Difference Of Gaussians.
LR	Learning Rate.
MC	Momentum Constant.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Object detection is related to computer vision and image processing that finds instances of real world objects of a certain class (such as humans, buildings, vehicles, etc) from image and videos. It has applications in many areas like image retrieval, security, surveillance and automated systems.

A video surveillance system provides visual surveillance in which observer is not directly on site. Surveillance performed directly and may stored and evaluated when necessary. It has many applications in security technology like alarm alert system(intrusion, fire), monitoring operations, loss prevention, event surveillance, parking lots, public areas etc. The objective of object detection and video surveillance system is to find objects(multiple objects) or alert the observer about the certain event from the video.

1.1 Background

Object detection and classification is an important problem in the field of computer vision. Object detection locate the objects in frames of video. The high power computers, the availability of high quality video surveillance camera increases the need for automated video surveillance applications. There are three key steps in video analysis for object detection and classification:

1. Detection and segmentation of interested moving objects
2. Extracting features of segmented objects
3. Classification of objects

Detection and classification of objects is complex due to:

1. Noise in image,
2. Complex object motion,
3. Partial and full object occlusions,
4. Change in illumination,
5. Real time processing requirements etc.

Some the questions that aeries for this problem definitions are:

1. How to find the interested object from all the moving objects?
2. How to separate similar looking objects from each other?
3. What features need to be extracted from interested object that efficiently classify the objects from each other?
4. Which classifier is accurate?

A large number of detection and classification techniques is available that answer these questions and mainly it depends on in which type of application we are interested.

1.2 Motivation for this project

Understanding the moving object in a scene by the use of video is a very challenging task. Also human operator systems are highly time consuming and less efficient. Thus it draws attentions of researchers, institutions and commercial companies.

For the traffic surveillance, vehicle detection and counting is very important problem. In large metropolitan areas, there is a need to keep details about vehicles in many cases. Here objects are defined as vehicles moving on road. 2-wheelers, 4-wheelers and heavy vehicles can be differentiated and vehicles are counted. To perform this task in real time that can able to identify the particular vehicle when came across camera is quite challenging. These were the main factors that motivated to study and design the current problem.

1.3 Objective of study

The objective behind this project is to detect and count the number of vehicles from the traffic video that are recorded by a stationary camera. And classify this vehicles into predefined classes such as 2-wheelers, 4-wheelers and heavy vehicles. And finally develop a system which is automatic, intelligent and able to perform well in real time also.

1.4 Scope of the work

To make a video surveillance system smart requires fast and robust algorithms for moving object detection and classification. The dataset from which features are extracted from various detected vehicles need to be efficient. This feature are also robust that can handle change in illumination, rotation, scale etc. efficiently. The classification on the testing data to its predicted class is also done precisely. The result is classification of vehicles in test video.

1.5 Outline of Thesis

The thesis will address four fundamental task for visual perception system, which are moving object detection, segmentation, feature extraction and classification. The thesis is organized as follows:

1. Chapter 2 reviews some most successful techniques for moving object detection and segmentation, feature extraction and classification.
2. In chapter 3 framework of the vehicle detection and classification is described and explained in detail.

3. Chapter 4 implements the vehicle detection, feature extraction and classification task. The system can identify vehicles and further it classify these vehicles into classes such as bus, truck, car etc.

4. Chapter 5 presents the summary of the thesis and suggestions for future work.

Chapter 2

Literature Survey

2.1 Object Detection and Segmentation Techniques

2.1.1 Frame Difference

Temporal or frame differencing is the most simplest and efficient approach. Frame differencing[1] is a difference between current frame and previous frame. This method will not identify the objects which remain stable for long period of time or uniformly colored objects. [2]

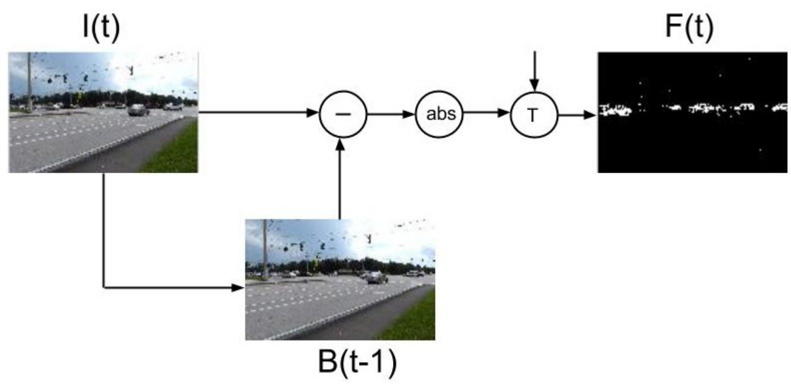


Figure 2.1: Temporal Differencing

$$BGS(i,j) = \begin{cases} 1 & \text{if } |BG(i,j) - I(i,j)| > thresh \\ 0 & \text{otherwise} \end{cases}$$

In this equation, input frame at time t is given. $BG(t-1)$ is the background frame at time $(t-1)$ and BGS is the result of background subtraction. For a given pixel at location (i, j), if the result of background subtraction was zero, this pixel is a part of scene; else, it is a pixel of moving object.

2.1.2 Background Subtraction

Detection of object of interest in the scene is the first stage in video surveillance systems. Background subtraction is a well known approach for detecting moving objects from video. An overview of various methods of background subtraction is provided in this section.[3]

A classification of background subtraction methods is given in [4, 5] by identifying two classes, namely recursive and non-recursive techniques. In recursive methods there is only one background model and it is updated with every new frame. In Non-recursive methods there is a buffer of n number of frames and prepaer a background model by the use of statistics.

In Table 2.1 a novel classification of state-of-the-art background modelling techniques is presented.

Buffered Based	Recursive	Non-Parametric	Fuzzy	BGS via Clustering
Mean Method	Approximated Median Filter	Kernel Density Approximation	Fuzzy Color Histograms	Codebook
Minimum maximum Filter	Single Gaussian	Condensation		K-means
Mediod Filter	Mixture Of Gaussian			
Eigen Backgrounds	Adaptive Mixture of Gaussian			

Table 2.1: Background Subtraction methods

2.1.2.1 Buffered Based Methods

Buffered based methods maintain a buffer L of video samples in order to compute the background by a statistical analysis of L.

1. **Mean Method** In this method background frame is obtained by using N frames and taking its average. background pixels remain present in most of the frames so by taking average of frames will give background frame. The current frame is subtracted from the average frame to get the foreground objects.[6]

$$BGS(i,j) = \begin{cases} 1 & \text{if } |BG(i,j) - I(i,j)| > thresh \\ 0 & \text{otherwise} \end{cases}$$

   Where, $BG(i,j) = \frac{\sum_{i=1}^{N} I(i,j)}{N}$

2. **Minimum Maximum Filter** The algorithm of this method operates only on grayscale video frames. Algorithm takes minimum difference and maximum difference and compare this difference to the predefined threshold. A pixel is in the background if:

$$|M_s - I_{s,t}| < \tau d_\mu OR |m_s - I_{s,t}| < \tau d_\mu \tag{2.1}$$

   where τ is an threshold which is defined by user and τd_μ is the mean of the largest frame difference. [7]

3. **Mediod Filter** In this method we can not find individually median of each color channel. Instead it can be estimates by the frames which is stored in the buffer[8]. The advantage of this method is finding the statistical dependencies between color channels.

4. **Eigenbackgrounds** In this technique we take N sample images and compute mean and covariance matrix. Each image is treated as vector after that mean is subtracted from each image and the average image is calculated. The average image is subtracted from each original image. Them Calculate eigen vectors and eigen values of covariance matrix. Eigen vectors are the directions in which the images differ from the mean image and the vector with the largest eigen value is kept.[9]

2.1.2.2 Non-Parametric Methods

In a typical non-parametric background model, background samples $Y = y_i,\ i \in [1, N]$ and foreground samples $Z = z_j, j \in [1, M]$ are maintained to compute the background and foreground distributions using a kernel function (typically a Gaussian function). Each pixel sample is encoded using its color and location,that is, $y_i = (x, y, r, g, b)$. The kernel-based background probability density is given by

$$p(x|B) = \frac{1}{N}\sum_{i=1}^{N} \varphi_H(x - y_i) \tag{2.2}$$

and the foreground probability is given by

$$p(x|F) = \alpha c + (1 - \alpha)\frac{1}{M}\sum_{j=1}^{M} \varphi_H(x - z_j) \tag{2.3}$$

where c is a constant (i.e., a uniform distribution), φ_H is the kernel function and α is a parameter which controls the relative contributions of the uniform distribution and the kernel function-based distribution [10].

1. **Kernel Density Estimation** This method is like a histogram. In order to smooth the curve we use gaussian PDF as kernel. [11]

$$P(I_{s,t}) = \frac{1}{N}\sum_{i=t-N}^{t-1} K(I_{s,t}, I_{s,i}) \tag{2.4}$$

where K is a gaussian kernel in this case and N is the total number of previous frames that is used to estimate probability. When color video frames is used, products of 1D kernels is used:

$$P(I_{s,t}) = \frac{1}{N}\Pi_{j=R,G,B}K\left(\frac{({I_{s,t}}^j - {I_{s,i}}^j)}{\sigma_j}\right) \tag{2.5}$$

Each pixel is classified as background or foreground based on the threshold value.

If $P(I_t) < thresh$ then pixel is belongs to background otherwise it is a moving object pixel.

2.1.2.3 Recursive Methods

1. **Approximated Median Filter** In this method median is estimated by the following update equation: [12]

$$B_{t+1}^{c} = \begin{cases} B_t{}^c + 1 & \text{if } I_t{}^c > B_t{}^c \\ B_t{}^c - 1 & \text{if } I_t{}^c < B_t{}^c \\ B_t{}^c & \text{if } I_t{}^c = B_t{}^c \end{cases}$$

   Here after initializing the background frame if the pixel value of current frame is greater than the pixel value of background frame then background pixel is updated by incrementing 1 else it is decremented by subtracting 1. Drawback of this method is that it adapts slowly towards large change.

2. **Single Gaussian** In this method each pixel is approximated by gaussian kernel and if the difference between mean and current pixel value is exceeds the threshold value then it is classified as foreground pixels [13].

$$|I_t - B_t| > \tau \tag{2.6}$$

   where τ is a pre-defined threshold. The background is updated by the equation

$$B_t + 1 = \alpha I_t + (1 - \alpha)B_t \tag{2.7}$$

   where α is kept small.

3. **Mixture Of Gaussians** In this method image pixels are clustered as a gaussian PDF of R, G, B component [14]. The probability of observing the current pixel value x at time t at a particular pixel location is:

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(x_t - \mu_{i,t}, \sum_{i,t}) \tag{2.8}$$

   The value of K is set in between 3 to 5 [15].

   Then each distribution is updated using adaptive learning rate. Here it is assumed that Gaussian distribution with the highest weight and lowest variance represent the background.

   Distributions are stored in decreasing order and then first C distributions are choosen as background model. If the current pixel matches one of the first C Gaussian distribution it is classified as background else foreground. [14]

$$\sum_{i=t}^{C} > T \tag{2.9}$$

   where T is threshold, are accepted as background.

2.1.2.4 Fuzzy Based approach

(a) **Fuzzy Color Histograms** Here the RGB pixel values is quantized into m histogram bins. Then this m colors in the CIELab color space to C clusters using fuzzy c means technique. By conducting FCM clustering, membership values of a given pixel to all FCH bins is obtained. After this feature vector of the pixel is obtained by summing the membership values of neighbouring pixels. Then each pixel is classified into background or foreground based on if it matches the feature vector or not.[16]

2.1.2.5 BGS via Clustering

(a) **Codebook** In this technique values at each pixel is quantized into codebook. Because the values are quantized it is in compressed form. By this we can capture background variation for a long period of time. By using this algorithm we can reduce the memory. This algorithm works best for detection of compressed videos. Illumination problem is handle by this technique.[17]

(b) **K-Means** In this technique each pixel is model by group of K cluster. Incoming pixels are compared against cluster group. A matching cluster is defined to have a Mannattan distance between its centroid and incoming pixel below a threshold. If no matching cluster is found the cluster with the minimum weight is replaced by the new cluster having the incoming pixel as its centroid.[18]

BGS Method	Findings
Frame Differencing [19]	• Easiest method, perform well for static background, less computational load • If the object is motionless for more than one frame period, it will consider as part of background
Minimum-Maximum Filtering [7]	• Operates only on grayscale videos which results in a loss of information compared to color video sequences.
Approximate Median [6]	• Easy to implement, pretty fast, memory consuming
Single Gaussian [13]	• Easy to implement and use, very fast • Can not handle non-static background

Mixture Of Gaussian [15]	• relatively slow, memory requirement is intermediate • Cannot deal with sudden, drastic lighting changes • Initializing the gaussian is important
Kernel Density Approximation [11]	• Speed is intermediate, Provide good accuracy, Memory requirement is high
Eigen Background [9]	• Faster than MOG, Limited to grayscale images
Fuzzy Color Histogram [16]	• Ability of greatly attenuating color variations generated by background motions while still highlighting moving objects. • Effective method for background subtraction in dynamic texture scenes. • Accuracy is more. • Histograms are sensitive to noise.
Codebook [17]	• Fast algorithm but not as much as KDE • Capability of coping with illumination changes • Compressed & adaptive bbackground model that can capture structural background motion over a long period time under limited memory.
K-Means [18]	• Suitable for deployement in real time application • Does not differentiate between moving objects and their shadows • Shadows are detected as saperate moving objects.

Table 2.2: Comparison: BGS Methods

In this paper authors discover moving objects by demonstrating pixel grey level distribution with time. They introduced Gaussian mixture model and the decision rule they have created for classification of pixels into background or moving object categories. Decision of approximated mixture segment for a given pixel is performed by probability maximization. Markov regularization is proposed for smooth detection. This method performs in real time. In this paper there is still problem of illumination variation is there and they tested on only CIF video frames. Another problem is that we have to instate background model and backgrounds having quick and extensive variations can not be demonstrated precisely. The method performs well for object integrity and able to eliminate false detections.[20]

In this paper moving objects is detected by frame difference algorithm and region combination. Moving regions is carried out by frame difference having adaptive threshold. Region combination is carried out by closest separation. The algorithm is able to detect moving object automatic. Here they choose only R component of each frame to detect moving objects to reduce running time so we lose some information. The objects with slow motion is sometimes classified as foreground.[21]

In this paper background removal technique is presented. Here image is divided into a 4×4 pixel patches and then every patch is treated as model and this model is adapted with every new image. The coefficient vectors are obtained by DCT. The number of vectors which represent the model is not fixed. An object is detected as foreground only if its neighbours differ from the background in previous image. The presented system is robust in illumination changes and it has low computational complexity. Here problem is that they only consider low frequencies. Since higher frequencies give more precise detail about the image, it should not be ignored, this will add some noise.[22]

In this paper a new technique is introduced in which foreground is extracted by improved GMM and chromaticity-gradient background subtraction method. The learning rate is updated with respect to illumination change. Mean and variance of the distributions is updated when there is a matching of new point. Shadow removal algorithm is proposed which doesn't relay on features but the contour of the moving object to detect shadow. Pixel is classified as background if it matches any distribution otherwise it is classified as foreground. The algorithm eliminates the imapct of illumination changes and shadows. The slow moving objects is also detected. Here there a problem of background initialization. We need to initiate background model. [23]

In this paper detection of multiple objects under multiple cameras in real time is shown. Cameras are arranged in such a way that the entry and exit of the objects can be handled in video. Algorithm use frame difference for background subtraction and finds centroid features for object representation. Then the results is given to the tracking module. They use multiple camera because to this will expand area. This approach deals with less misdetection rate but the processing rate and recognition of detected object is slow. This algorithm is not used any classifier so it is not robust. [24]

In this paper moving objects is detected by improved background subtraction algorithm and feature based approach is introduced. Objects which are detected are counted by indexing. Median filter is used for removing noise and features like area, centroid and average of RGB pixels is used for tracking. Here the problem is still arise that we need to initiate the background model. [25]

2.2 Feature Extraction Techniques

Feature extraction means identifying the pattern of interest from the given image. It represent intersted parts within image. It can be used in detection, classification, recognition and pattern matching.

2.2.1 Extract features in local image neighborhoods

Here image is divided in the form of grid, from each region of the grid, features are extracted and feature vector are represented by its central pixel. This is useful for identifying structures and spatial information within an image.

This region features helps to describe:

- Local texture
- Local intensity distribution (histograms)
- Local gradient, edge strength and orientation
- Local appearance

2.2.1.1 Region Feature Types

(a) **Local mean and standard deviation**

In this techniques from each image block, mean and its standard deviation is calculated. This is operate on a single image channel. Here by default image is divided into $8x8$ image blocks and features are extracted from each block if it contains any information. If block contains any information then its mean is calculated, then for that block its standard deviation is calculated. So output contains two feature vectors: first contains mean vector and second contain standard deviation.

(b) **Local histograms**

In this technique, histogram is made for each region. This require specific range within the data and operates only on single color channel. Data range is defined with vector which contains minimum and maximum value of block. Values outside the range are discarded. Histogram can be obtained by classifying the pixel value with respact to data range. By default range is divided into 8 bins. From this histograms various features are extracted like mean, entropy.

(c) **Co-occurrence matrices**

This is a two dimensional matrix in which finds how often pixel with a gray-level value i occurs horizontally adjacent to a pixel with the value j. Each element (i, j) in the matrix represent the number of times that the pixel with value i occurred horizontally adjacent to a pixel with value j. This method is also work with single color channel. With this image is scaled. Then the feature vector is calculated.

2.2.2 Representing objects identified in an image

In this method, feature vector is calculated for each object that is identified in the image. Here first step is, from the original image region features are extracted.

Then classifier is trained to differentiate object and background. When new image comes, each pixel is declared as either object pixel or background pixel. Second step, connected components are segmented and it is called object. This object is represented by feature vectors.

2.2.2.1 Object features

(a) **Object Size** In this method size of the detected object is calculated by means of height and width. This size represent the feature vector. This helps in classifying the objects by its size.

(b) **Mean of object pixels** In this method mean vector is calculated for each detected object. Mean vector represent feature vector. This can be used to handle uniform types of objects.

(c) **Sum of object pixels** In this method feature vector is the vector which can be computed by sum over each object.

2.2.3 Transform color space

In this method, input is a pixel in a color image and output is a pixel in a different color space. This technique can be used for color based object segmentation.

2.2.4 Haar Features

Haar features are similar to convolution kernels which are used to detect the presence of that feature in the given image. Each feature results in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle.

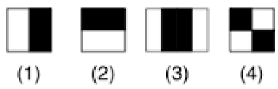


Figure 2.2: Basic Haar like features

Basically algorithm uses 24×24 window as the base window size to start evaluating features in any given image so it will end up by calculating about 160000+ features. So we have to eliminate lot of feature which are not useful or redundant and select those that are useful to us. This is done by adaboost.

Pros	Cons
Uses image region differences	Misses texture, shape information
Very fast feature evaluation	Sensitive to image intensity
Slow training time	Not good for general object detection (like people)

2.2.5 Histogram Of Gradients

HOG was introduced by Dalal and Triggs. Image is taken as a grid of cells. From each cell histogram is computed over orientation bins. At each pixel, image gradient vector is calculated. The angle of the vector is used as a vote into the corresponding orientation bin and the vote is weighted by the gradient magnitude. Votes are accumulated over the pixels of each cell as shown in Figure 2. The cells are grouped into blocks and a robust normalization process (HOG normalization) is run on each block.

The normalized histograms of all blocks are concatenated to give the window-level visual descriptor vector for learning. The blocks overlap spatially so that each cell appears several times with different normalizations, as this typically improves performance.

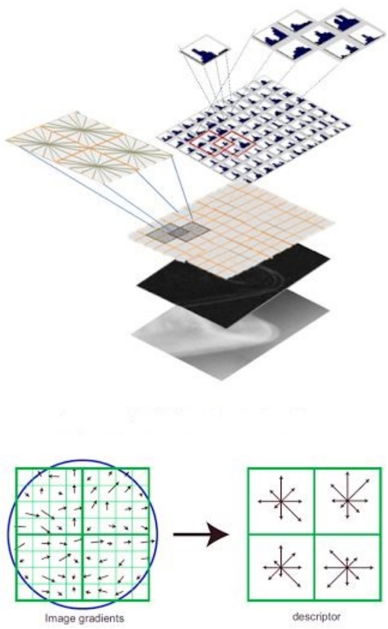


Figure 2.3: HOG features

2.2.6 Edge detection

An edge can be defined as a place of rapid change in the image intensity. The points at which image brightness changes sharply are organized into set of curved line segments called edges. This can be computed by taking partial derivative of the image with respect to convolution function.

2.2.7 Centroid

Centroid is also called center of gravity of the shape. Its position should be fixed in relation to the shape. Centroid is calculated by the following equation:

$G_x = \frac{1}{N}\sum_{i=1}^{N} x_i$

$G_y = \frac{1}{N}\sum_{i=1}^{N} y_i$

where N is the number of point in the shape.

2.2.8 Complex Coordinates

This is calculated from the value of boundary points.

$z(n) = [x(n) - g_x] + i[y(n) - g_y]$
where (g_x, g_y) is the centroid of the object.

2.2.9 Centroid distance function

This feature vector is calculated by finding the distance between boundary points and centroid (g_x, g_y) of the object.

$r(n) = [(x(n) - g_x)^2 + (y(n) - g_y)^2]^{\frac{1}{2}}$

2.2.10 Area function

This can be calculated by finding the area between two points of the object boundary from its centre. This is shown below in the figure.

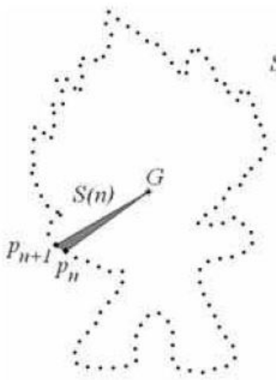


Figure 2.4: Area function

2.2.11 Scale Invariant Feature Transform

This features are used when object changes its scale. For the feature detection, points are detected that are repeatably selected when scale or location changes. These points are called keypoints (also known as feature points). Then orientation is assigned to each keypoints. This features are useful in many applications.

2.2.12 Appearance Based approach

(a) **Principal Component Analysis** This technique is used to reduced the amount of data or features. When there are large amount of data or features, out of them some are redundant. It is not efficient to use that because it will take more memory, more computation and time. To reduce this kind of features in size PCA is used.

2.3 Object Classification Approaches

1. **Shape based classification**
   In this method classification is performed by using the shape information of moving object.

   (a) **Using Aspect Ratio**
   Once the moving objects are identified, noise are removed from that and some morphological operations are applied on that. Objects which have very small regions are discarded by considering area. Features like bounding box, area, mean-value of detected region are calculated. By using these features classification is performed. Aspect ratio is calculated by ratio of height to the width.
   Remarks:
      i. Unwanted background pixels may be detected as foreground pixels.
      ii. False detection may happen.

   (b) **Silhouette based method:**
   Silhouette contains detailed information about the shape of object. From silhouette various features are extracted like corners of object, histograms, aspect ratio, disperseness. By using these features classification is performed.
   Remarks:
      i. Some background may look like objects. Or some objects look like other objects.
      ii. If the object is temporary occluded, it will not affect in the classification.

2. **Decision Tree**
   Classification using decision tree is a one of the simplest method. In this method data-set are partitioned into number of subsets. In this tree, each node contain conditions that separate features or records that have different characteristics and accordingly decisions are made. Once the decision tree is made, classification of test data is done very easily. First at the root node condition is checked and then following the appropriate branch as per outcome of the test value. This process is continued until leaf node is reached. Class label associated with the leaf node is assigned to test data value. Like-wise class labels are predicted from the test data.
   Remarks:

   (a) Constructing optimal decision tree is main problem in decision tree classifier because of the exponential size of the search space.
   (b) Construction of decision tree is computationally inexpensive.
   (c) Once tree is made, classification of test data is extremely fast.

3. **Support Vector Machine (SVM)**
   Generally support vector machine is used when data-set have exactly two class to classify data. But now multiclass classification is also performed using SVM. To separate the data between two class, best hyperplane is used by the SVM classifier. This best hyperplane is obtained by finding the maximum distance between two classes, which have no interior data points. Here data for training is vectors with their class. Then by using some mathematical equations best hyperplane is calculated and test data is classified by comparing its value with the hyperplane.
   Remarks:

(a) Performance and accuracy is depends upon hyperplane selection.

4. **Artificial Neural Network (ANN)**
ANN is system based on person mind, which contains large number of neurons and each neuron is also connected to other neuron/neurons. Neural network can be used to classify static patterns. When network are trained using supervision it is known as perception. These network require labelled training data. From this information, output is calculated. Here each neurons contains some weight with it and decision are made by using decision boundary. This boundary is obtained by calculating mathematical function and test data value is compared with this boundary. If test value is fall within certain region then it is considered to be in class represented by that region. The number of internal nodes/neurons within input and output neuron layers is called hidden layer. An error is calculated by the difference between response and the system output. This error is fed back to the system and system weights are adjusted. This process is repeated until the performance is acceptable.
Remarks:
(a) Efficient in terms of time and resources.
(b) Here designer choose network topology but system adjust parameters automatically so it is difficult to bring priory information into design.
(c) Performance and accuracy depends on network structure and parameters.

5. **Bayesian network classifiers:**
(a) **Nave bayes:**
Nave bayes classification is based on bayes theorem. When dimensionality of input is high this method is used. In this method, features are assumed to be independent given class. This method will classify the data with the highest probability. Training data set contains labelled data. Then probability of each features is calculated for each class. When test data comes its probability is calculated with respect to each class and which is highest probability that value is predicted to that class.
Remarks:
i. This method is fast and space efficient.
ii. Fast to train and classify.
iii. Not sensitive to irrelevant features.
iv. Assume independence of features.
(b) **Bayesian belief network:**
BNN is a graph which contains probabilistic dependencies. This graph contains interconnected nodes. Node represent variable and arc represent relationship between variable. This will work by calculating joint probability and based on bayes theorem. Network provides graphical relations on which learning can be performed. We have set of training data and information about each data that is belongs to which class. Then conditional probabilities are estimated, these values shows what is the probability that value or feature is belong to given class. To make the decision about new test data probability of that value belongs to some class is determined by conditional way. And whichever has

the highest probability test data is classified into that class.
Remarks:

i. Different network can be made for same problem.
ii. Computationally infeasible.

Chapter 3

Vehicle Detection and Classification for Video Surveillance

3.1 Introduction

Nowadays, every public places and highways are monitored using video surveillance camera. Traffic is increases day by day. With the help of traffic surveillance camera that are installed on highways live monitoring of the traffic scenarios is possible. The operator of traffic management centre needs to observe from the video. This is very difficult and time consuming and sometimes becomes boring, and miss-detection can be happen by them. So there is a need to automatize the tasks such as the detection of dangerous situation, license plate recognition, information about traffic flow, recognition of vehicles that passes from that highways, finding the vehicles that is in over speed, license plate recognition, information about accident, traffic congestion etc. Vision based traffic surveillance system helps to alert about these situations. There is no need to observe for long period of time. The tasks are carried out completely automatic. Vehicle counting and classification are important factors for finding the number of vehicles and what percentage of their classes run on a particular road. This information helps relevant authorities.

In this application, we assume that there is single static camera that is mounted on the highways. The system is able to detect vehicles as they move through cameras view and classify each individual object in several categories.

3.2 Approach followed

Problem Description: Prepare data-set for traffic surveillance, detect and segment the vehicles from video (if possible in real time also), classify vehicles into predefined class and also count number of vehicles.

1. **Task 1:**Prepare and collecting the data-set for traffic surveillance system. Data-set should be in good quality and contain useful information.

2. **Task 2:**Finding the multiple moving objects in video frames.

3. **Task 3:**Detecting and segmenting only vehicles from video frames. And arrange this vehicles according to predefined classes.

4. **Task 4:**Extract the features from the segmented vehicles according to its class.

5. **Task 5:** From these features system can learn about vehicle class using machine learning algorithm and classify the new data efficiently.

3.3 Generic Framework of the System

A generic framework for object detection and video surveillance is shown in fig. **??**. The majority applications for surveillance are organized in a hierarchical way, with low level image processing techniques feeding into tracking algorithm which in turn feed into higher level scene analysis and/or behaviour analysis module.

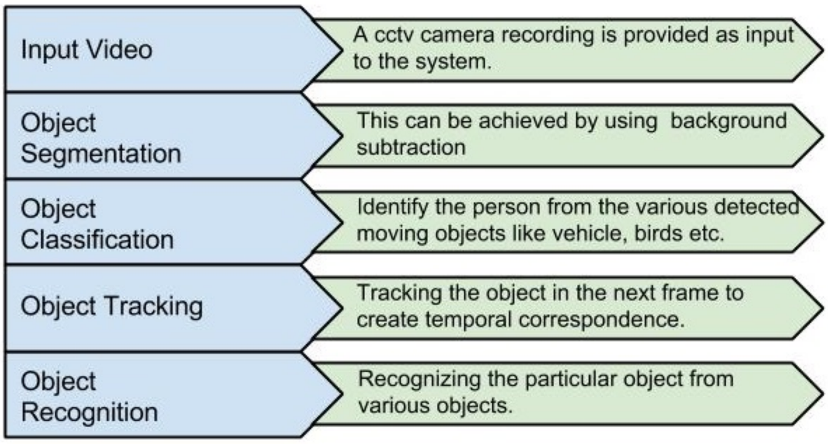


Figure 3.1: Generic Framework
[26]

There are three key steps in video analysis: Detection of interesting multiple moving objects, tracking of such objects from frame to frame and analysis of object tracks to recognise their behaviour.

Moving object detection is the basic step for further analysis of video. It handles segmentation of moving objects from stationary background objects. Commonly used techniques for object detection are background subtraction, point detection, segmentation, supervised learning techniques, temporal differencing. Due to dynamic environmental conditions such as illumination changes, shadows and reflection, waving tree branches in the wind and changes in appearance and shape of object is difficult and significant problem that needs to be handled well for a robust video surveillance system.

Object classification step categories detected object into predefined classes such as background objects or foreground objects. It is necessary to distinguish interested object(s) from other objects in order to track them.

The next step is to track the detected object frame to frame. This gives useful information about the object. The output generated by tracking step is generally for object classification and higher level activity analysis. The final step is to recognize the tracked object.

3.4 Framework for defined system

Detection of vehicles can be implemented by different methods. Here main problem is changing in light and traffic flows. In proposed system, traffic video(recorded/live) is given as a input to the system and by converting it into frames, background can be extracted and then detection of vehicle is performed.

The system has following stages:

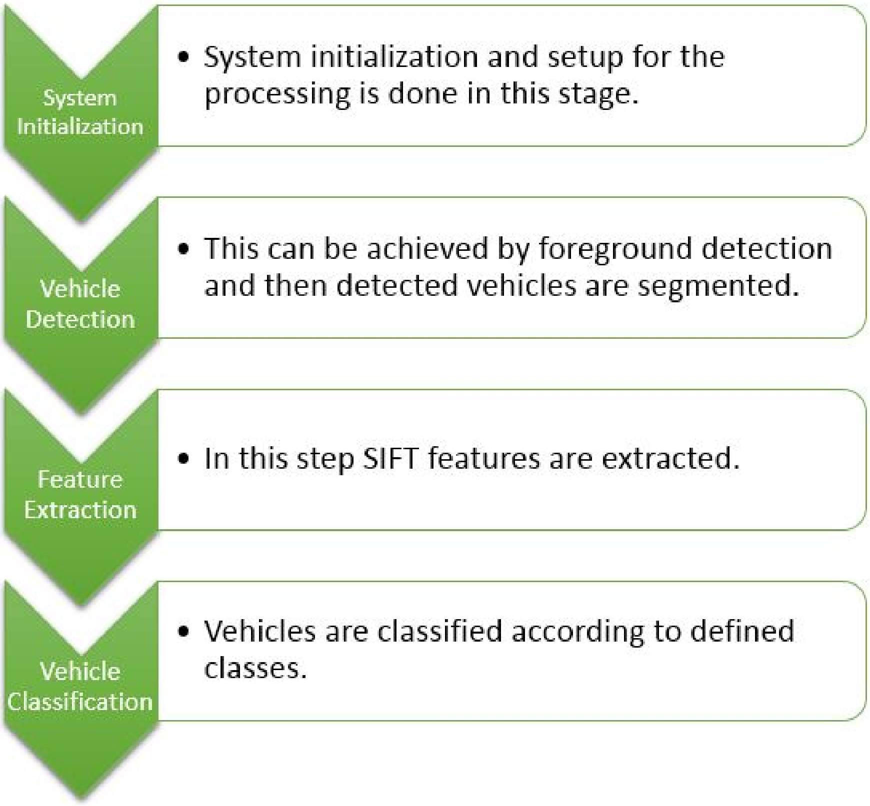


Figure 3.2: System Framework
[26]

1. **System Initialization:** System initialization and setup for the processing is done in this stage. If recorded video is given as input then the frames extracted from it is given as a further analysis or if camera records a live video then that continuous data frames is send to the system for further analysis.

2. **Vehicle Detection:** In this step first moving objects are identify by finding the foreground using background subtraction method. Then after applying some operations Vehicles are detected and segmented. This process is explained in detail later.

3. **Feature Extraction:** In this step, features are extracted from the detected vehicles. Here we compute SIFT features from each segmented vehicles. The output of this step is applied to classifier in the next step.

4. **Vehicle Classification:** In this step, vehicles are classified using neural network approach. This step is also explained in detail later.

3.4.1 Literature Survey:

Paper	Detection Method	Feature used	Classifier used	Classified objects into	Remarks
[27]	BS (GMM)	Edge and corner	SVM classifier	Vehicles and non-vehicles	System is not depend on camera height, vehicle size.
[28]	BS	Width and height	Classification based on vehicle dimension	Cars and non-cars	• Provides location and velocity information for each vehicle. • Does not handle shadows, illumination and weather conditions. • Classification is not accurate.
[29]	Gradient based detection	-	Adaboost Classifier	Vehicles and non-Vehicles	• System work in real-time and false positive rate is very low. • raining data set contains only car images.
[30]	Active Learning	Haar like Features	Adaboost	Vehicles and non-Vehicles	• System has high precision, recall. • New performance metric is defined.

[31]	Visual Background Extraction	HoG	SVM Classifier	Car, Motorcycle, lorry, background (without vehicles)	• Provides good accuracy. • Features used in this paper generally used for human detection.
[32]	Active Learning	HoG, Haar-like features	SVM, Adaboost	Vehicles and non-Vehicles	
[33]	ROI	Area	Area is greater then mean area	Car, Truck	Does not work well for high density traffic.

Table 3.1: Approaches used in different Vehicle Detection and Classification Systems.

3.5 Problems and assumption in Vehicle Detection and Classification

3.5.1 Problems identified

1. **Occlusion:**This is an important problem in vehicle detection even if the camera resolution is high. Obtaining individual vehicles from blobs are difficult when they are occluded.

2. **Scale in change:** When vehicles enteres in the camera frame and when its leaves the camera frame its scale changes. So we need to taken care of this problem.

3. **Illumination problem:**In the night time, in the difficult light, or in the different varying environment conditions like (rain, fog, etc.) it is very difficult to detect and differentiate between vehicle class.

4. **Large variations within the vehicle class:**Vehicle contains large variations of class within it like heavy vehicles, light weight vehicles, two wheelers, four wheelers, low speed vehicles, high speed vehicles, long and short vehicles and many more. Sometimes its difficult to separate vehicles within these classes.

5. **Diversity of camera views:**This is very important issue that needs to be taken in consideration because different places has different camera views. So algorithm should work in all conditions.

6. **Resolution of camera:** When camera resolution is high then we can extract more information from it. When its resolution is low it contain less information. So sometimes this thing cause problem.

3.5.2 Assumptions

1. Here we assume that there is single camera and it is static.

2. Traffic videos contains vehicles only. Other types of objects are very less in the video or not present in the video.

3. Various parameters and values are taken as per given in the research paper. And some of parameters are decided by the experimental way.

4. Area of detected blob having area less than 150 are discarded.

3.6 Dataset Preparation

Dataset is prepared for this application. Some of the videos in the dataset are recorded manually, some are taken from the surveillance camera, some are available dataset videos and some are taken from the internet. The videos are of traffic which contains various kind of vehicles and other kind of objects are very limited. The videos are taken and recorded in different timings of day. And camera situation in all the videos are different. So dataset is well suited for any kind of traffic surveillance application. Here we assumed that camera is static.

- The following are the video list that are taken from the internet source.
  1. ACTi KCM-5611 - Traffic Surveillance (32 seconds, 1280x714 frame size) [34]
  2. Austin Texas Video clips traffic (3 minutes 2 sec, 1280x720 frame size) [34]
  3. Busy traffic of Kolkata - West Benga (1 minutes 43 sec, 1280x720 frame size)[34]
  4. IP Camera Road Traffic Surveillance Demo Day (40 seconds, 1280x720 frame size) [34]
  5. KCM-7111 Outdoor Day Bus Stop (18 seconds, 1280x720) [34]
  6. KCM-7111 Outdoor Day Traffic Intersection (17 seconds, 1280x720)
  7. M6 Motorway Traffic (34 minutes 8 sec, 1280x720) [34]
  8. Static shot of bus stop with busy traffic (26 seconds, 1280x720) [34]

- The following videos are taken from the surveillance camera.
  1. Main Gate01 (59 minutes 59 sec, 704x576 frame size) [34]
  2. Main Gate02 (59 minutes 3 sec, 352x288 frame size) [34]

- The following videos are recorded manually.
  1. TrafficVideo1 (2 minutes 20 sec, 1920x1080 frame size) [34]
  2. TrafficVideo2 (59 sec, 1920x1080 frame size) [34]
  3. TrafficVideo3 (45 sec, 1920x1080 frame size) [34]

- The following videos are available dataset for traffic surveillance applications. LISA video dataset Description: [34] This video dataset contains color video sequences captured at different times of the day (Morning, evening, sunny, cloudy, etc.), different driving environment, Varying traffic conditions.
  1. Video 1: 33 seconds, 750 frames, 320x240 frame size
  2. Video 2: 30 seconds, 500 frames, 320x240 frame size
  3. Video 3: 8 seconds, 120 frames, 160x120 frame size

3.7 Vehicle Detection and Segmentation

MATLAB has been used for performing the vehicle segmentation and post processing steps to remove noise using user defined as well as built-in functions provided by the image processing and computer vision toolbox.

Flow chart of this step is shown below:

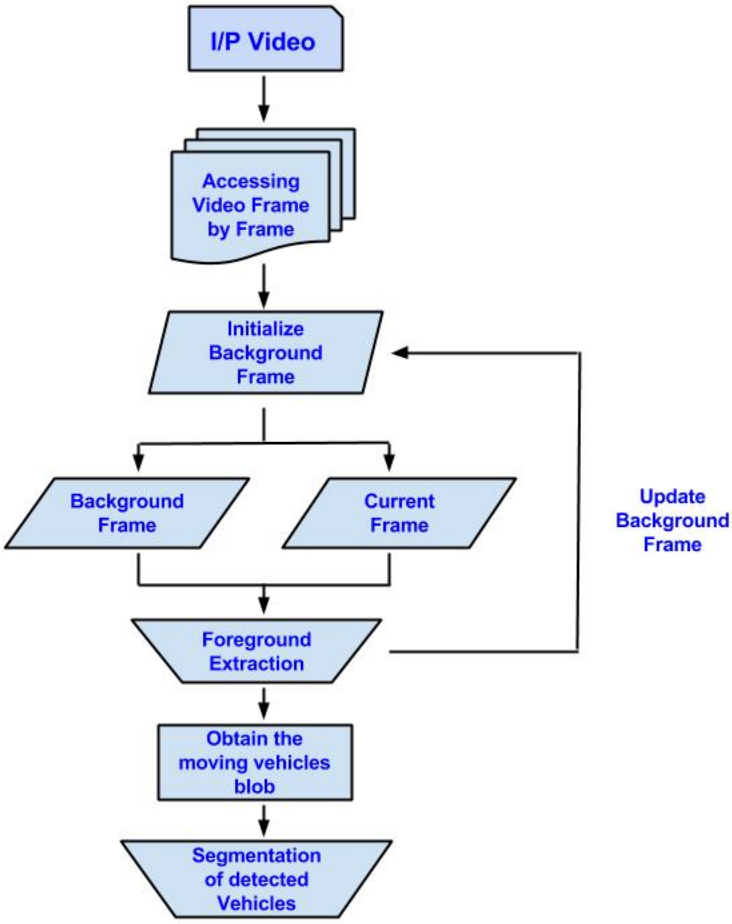


Figure 3.3: Flow Chart of Vehicle detection and segmentation

Here we have video frames as a input. And for finding the moving object, frame difference algorithm is used. Here first frame is taken as background frame. Threshold value is decided by the experimental way for each video. Then by using frame difference with given threshold value foreground are extracted.

From foreground pixels, blobs are detected. Here we take blobs which have area bigger then 150. So vehicles are identify easily and every blobs having area less than 150 are discarded. When vehicles are leaving or entering in the video frame, depending upon the camera situation, it will became smaller and smaller in size. So its area is reduced and becomes very small. And sometimes blobs with small area are noise in the frame. So it is better to discard such blobs.

Out of all moving objects for finding the vehicles in the video frame, square objects are identify using structural property because vehicles are look like square when blobs are detected. Then morphological operations are applied to the detected vehicles and this detected vehicles are segmented from the frame.

After segmentation of vehicles, these vehicles are arranged according to various classes within vehicle class.

In this step from all the videos we get 5180 segmented vehicles in different views like side view, front view and rear view. These are stored in terms of images and have different classes within vehicle class. Classes are mentioned in the classification step.

3.8 Feature Extraction

Here SIFT feature is used because various issues that came across while developing this application. The major issues are as follows:

1. Illumination problem
2. Change in scale
3. Occlusion

SIFT is invariant to rotation and scaling and partially invariant to change in illumination, camera viewpoint, occlusion. Hence this features are best suited for the application. [35]

SIFT contains two parts: 1) SIFT detector and 2) SIFT descriptor and these features are 128 dimensional vector. The following figure shows the flow chart of this step:

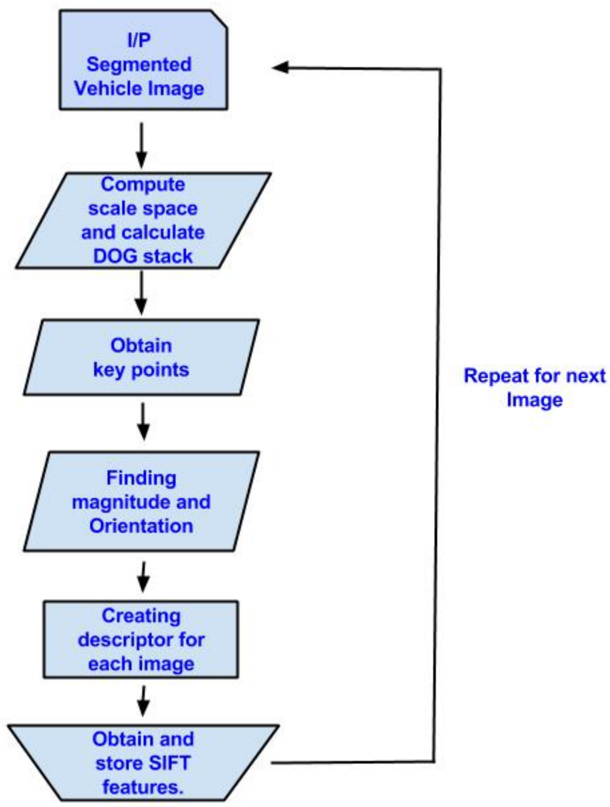


Figure 3.4: Flow chart for feature extraction

Image scale space stack is obtained by convolution of the image with gaussian kernel. After obtaining scale space, DOG is calculated and finally image stack is made.

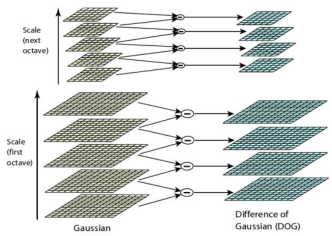


Figure 3.5: SIFT - Scale Space

After obtaining scale space stack, each pixel value of the image is compared with the 26 neighbours of it. If current pixel value is greater or less than all the 26 pixel value then current pixel is known as keypoint. This keypoint is stored this are generally not changed when image scaling occur or rotates. Hence SIFT is invariant feature.

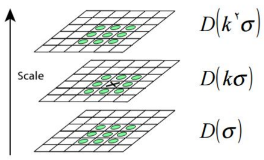


Figure 3.6: Extrema Calculation

After calculating keypoints, gradient and its orientation are calculated. Orientation are represented by histograms. Then magnitude and orientations are sampled around keypoint location. Size of image gradient is 4x4 and for each sample histoogram in 8 direction is calculated. So we get 4x4x8=128 feature vector for one image.

3.9 Vehicle Classification

The class in which vehicles are classified is listed below:

1. Bike
2. Bus
3. Car
4. Mini Truck
5. Rickshaw
6. Truck

Classification approach used for this application is shown below:

In this application vehicles are classified using neural network. Neural network is made by input nodes, hidden layer and output node that are interconnected together.

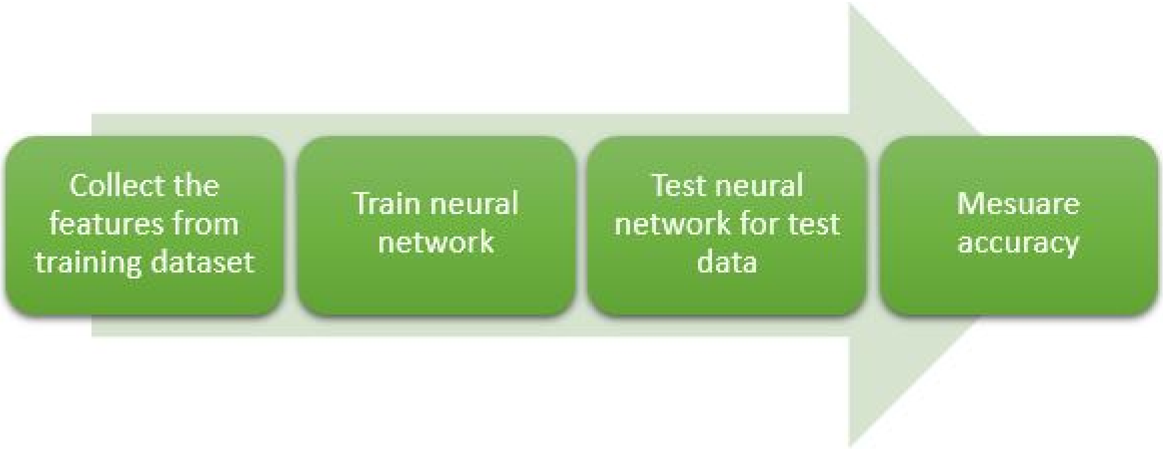


Figure 3.7: Flow chart for Vehicle Classification

Each node is assigned weight and function.

There are different types of neural network available but they are basically classified into types: [36, 37]

1. **Feed-forward network** In feed forward neural network, features or data are applied as a input to the network and this values are passes layer by layer until the output reached. There is no data transfer between layer in this type of network. Hence it is called feed-forward network.

2. **Feed-back network** In feed back neural network, data travelling in both directions and this process is continues until it reaches in desired state. This type of network can be used when optimization problems occures.

When we have linear data that are easily separable than there is no need for hidden layer in neural network to classify them. These data are easily classify by finding the best line that separate it. But when we have data which are not linearly separable, then we need hidden layer in neural network. These type of data are given as a input to the network along with some weight and function and output is obtained.

Neural network has two phase:

1. **Learning Phase:** In the learning phase weights are modified in such a way that data are classified into correct category. When input reaches to the output node, output is compared with the actual output and if its not matched then weights are modified. If this procedure is performed for every pattern and category pair then 1 epoch of learning is performed. This phase will depend upon the size of network, number of patterns to be learned, number of epochs, speed of computer.

2. **Classification Phase:** In classification phase, output which are obtained on the output nodes are analysed. Classification is performed by selecting the class associated with the highest output value.

Number of hidden neurons in the hidden layer is decided by the following equation:

$No_of_hidden_neurons = \frac{Noofhiddenlayer+Noofinput}{Noofhiddenlayer}$

If we take more number of hidden layers then equations with more power is generated and its value becomes almost zero. So 2 or 3 hidden layer is well suited for classification.

There are many type of input functions is available to train the network like gradient descent, gradient descent with momentum, gradient descent with momentum and adaptive learning rate, sigmoid functions etc. Here we take gradient descent with momentum function.

Gradient descent with momentum allows network to react when minima occur. It will slide through this minima with the help of momentum. This function works with two parameters that are learning rate lr and momentum constant mc. Value of lr and mc is in between 0 and 1.

Training is continued till input,weights and function has derivative function. Backpropagation is used to calculate derivatives of performance $perf$ with respect to the weight and bias variables X. Each variable is adjusted according to gradient descent with momentum,

$dX = mc * dXprev + lr * (1 - mc) * dperf/dX$

where $dXprev$ is the previous change to the weight or bias.

Chapter 4

Implementation Results

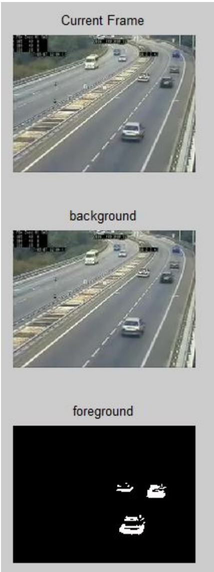
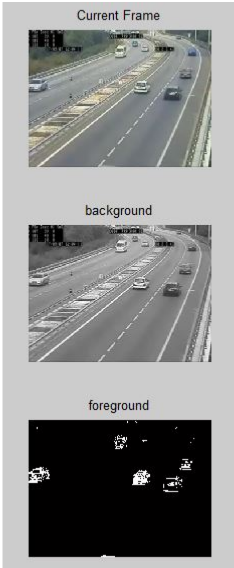
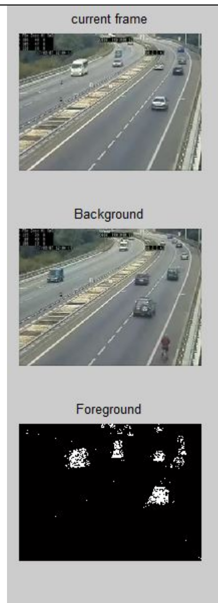
4.1 Vehicle detection and classification using existing technique

4.1.1 Foreground Extraction from video

Algorithm 1 Algorithm for Background Subtraction

Inputs: Video frames in binary format. Number of frames in videos. Width and height indicating size of frames.
Output: An image showing the moving object.
$thresh = 25$
Initialize background frame.
for $i = 2 : nFrame$ **do**
    Read the frame and convert it into binary.
    Find the frame difference
    for $j = 1 : width$ **do**
        for $k = 1 : height$ **do**
            if $FrameDifference > thresh$ **then**
                $FD[k, j] = frame(k, j)$
            else
                $FD[k, j] = 0$
    Update background frame.

4.1.1.1 Snapshots

Frame Difference	AMS	MOG
		

4.1.1.2 Analysis

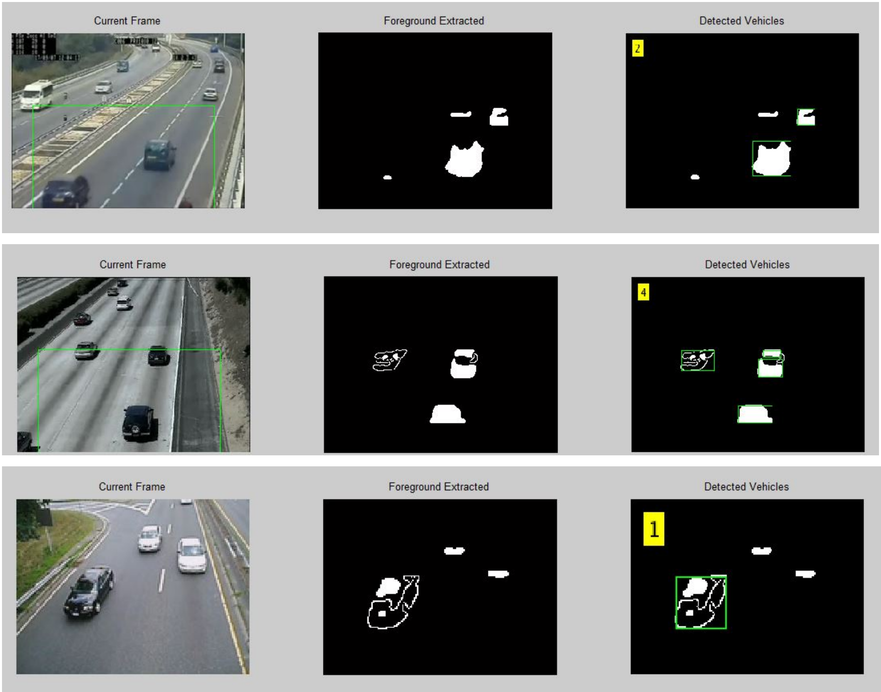
Input Video	No. of frames	Fps	Accuracy		
			FD	AMS	MOG
Video 1	750	25	97.92	93.57	98.86
Video 2	500	15	92.07	90.52	96.57
Video 3	120	15	99.24	92.87	99.59

4.1.2 Detection of objects and vehicle counting

Algorithm 2 Algorithm for vehicle detection and counting

Inputs: Foreground frames.
Output: Detected vehicles and number of vehicles within frame.
$K = Canny\_EdgeDetection(foreground\_frame)$
$G = MorphologicalOpening(K)$
Measure the height and width of detected region
Fill the hole in regions and calculate aspect ratio.
Bounding box is formed around the detected objects.
Calculate number of bounding boxes within frame.

4.1.2.1 Snapshots



4.1.2.2 Analysis

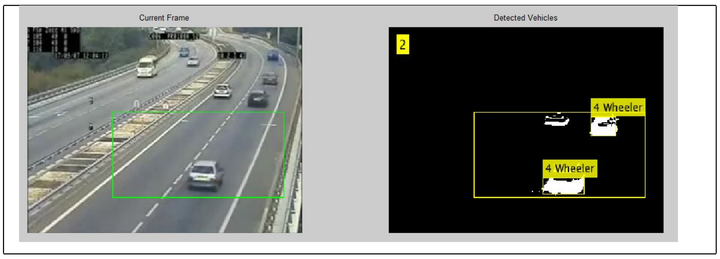
Input Video	Format	Actual moving objects	Detected moving objects	Accuracy
Video 1	RGB	33	31	93.33%
Video 2	RGB	43	40	93.02%
Video 3	RGB	10	10	100%

4.1.3 Classification of vehicles

Algorithm 3 Algorithm for classification of vehicles as 2, 4-wheeler and heavy vehicle

Inputs: Detected objects(vehicles)
Output: Objects with its type
for each detected objects **do**
    Shape based Feature extraction (calculate area of detected objects)
    Store features individually
Define classes according to features.
Perform classification.
Test on testing dataset

4.1.3.1 Snapshots



4.1.3.2 Analysis

Input Video	Total no. of 2-wheeler	Total no. of 4-wheeler	Total no. of heavy vehicles	Correctly Classified	Incorrectly Classified	Turn-around time (in sec)
Video 1	1	31	1	30	3	178.515 sec
Video 2	0	42	1	42	1	110.613 sec
Video 3	0	10	0	6	4	14.94 sec

4.1.4 Observation and Limitation of approach

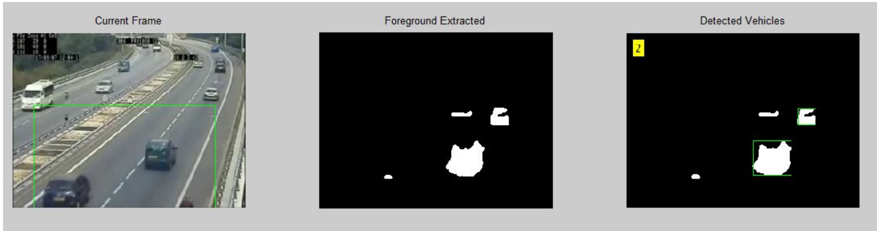
1. This approach works well for videos taken from rear.
2. Detection rate and classification rate is good.
3. Limitation of the approach is that sometimes it also detect pedestrian as vehicles.
4. There is need for finding more accurate features of vehicle.
5. Make the algorithm such that it will work precisely for the videos taken from the rear, front or side end.

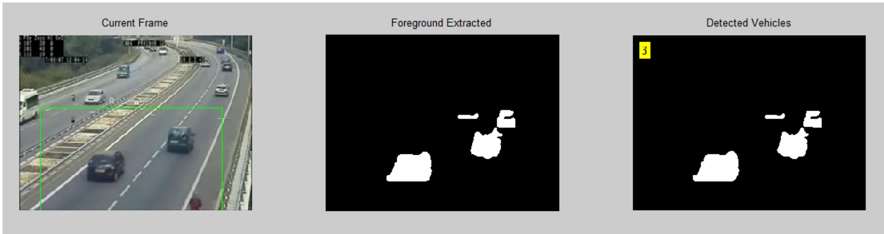
4.2 Vehicle Detection and classification using proposed approach

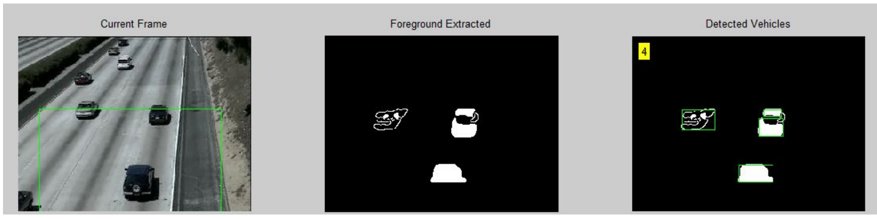
4.2.1 Vehicle Detection and Segmentation

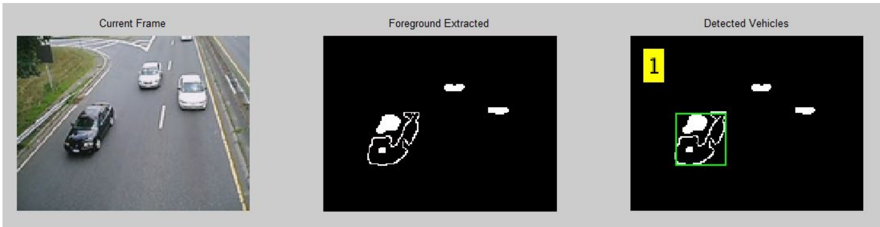
MATLAB has been used for performing the vehicle detection and segmentation of vehicles from other objects in the video.[38] Flowchart of vehicles is shown in figure 3.3. Here it is assumed that camera is static. To extract the background from foreground frame difference backgroun subtraction method is used. For removing noise from the foreground median filter is applied. Here various types of vehicles is segmented from ongoing video and stored as an image. There are total 6097 vehicle image is obtained. This dataset contains 1109 Motor Bike images, 150 Bus images, 4009 Car images, 453 Mini Truck images, 16 Auto rickshaw images and 360 truck images. The results of this step is shown in the table.

Vehicle Detection









Vehicle Segmentation

1. **Bike**

2. **Auto Rickshaw**

3. **Bus**

4. **Car**

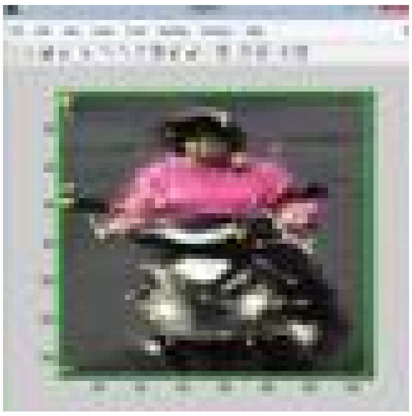
5. **Mini Truck**

6. **Truck**

4.2.2 Feature Extraction

In this step features are extracted from the segmented vehicle image. From each image SIFT feature vector is obtained. SIFT is an 128 feature vector. Keypoints are extracted and descriptor is obtained for each image. Each image contain different value of descriptors. So feature vector for whole data set is made by taking the highest value of descriptors. Other values are appended as zero.

Results of this step is shown below with descriptors:



4.2.3 Classification of Vehicles

Vehicle classification is performed using neural network approach. Vehicles are classified into 6 classes that are bus, bike, car, mini truck, truck and auto rickshaw. They are classified using the features obtained from the above step. Here number of hidden layer is 2, number of output is layer is 6 and number of inputs to the neural network is number of features obtained.

The result of this step is shown below:

1. Neural Network with 1 Hidden Layer
   Learning rate = 0.01, Momentum Constant = 0.9

P / A	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	58	-	4.5	9.5	28	-
Bus	-	60	7	13.5	0.5	19
Car	2	10	54	30	3	1
Mini Truck/van	-	12	28	57	2	1
Auto Rickshaw	23	5	9	8	51	4
Truck	-	35	12	3	2	48

2. Neural Network with 2 Hidden Layer
   Learning rate = 0.01, Momentum Constant = 0.9

P / A	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	63	-	4	8	23	2
Bus	-	68	5.3	11.7	0.8	14.2
Car	0.8	10.2	63	22	2.8	1.2
Mini Truck/van	0.2	13.4	22.6	63	0.5	0.3
Auto Rickshaw	19.8	7.6	6.4	5.83	54	6.36
Truck	-	29.6	11.4	1.83	2.3	54.8

3. Neural Network with 1 Hidden Layer
   Learning rate = 0.05, Momentum Constant = 0.9

P / A	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	65.8	-	3.9	9.5	20.8	-
Bus	-	69.3	3.4	7.6	3.6	16
Car	2.8	9.2	59.3	26.6	-	2.1
Mini Truck/van	-	8.3	20.5	62.4	4.7	4.1
Auto Rickshaw	16.2	3	9.3	6.8	93.3	6.7
Truck	-	28.1	13.2	1.8	1.9	55

4. Neural Network with 2 Hidden Layer
   Learning rate = 0.05, Momentum Constant = 0.9

P / A	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	61.2	-	3.5	7.8	24.8	2.7
Bus	-	65.6	5.8	14.2	1.8	12.6
Car	3.8	7.9	61.3	26.7	0.28	0.02
Mini Truck/van	-	14.8	20.7	60.08	3.6	1.22
Auto Rickshaw	13.8	3.6	7.9	12.2	58	4.9
Truck	-	23.6	14.8	4.6	3.8	53.2

5. Neural Network with 1 Hidden Layer
   Learning rate = 0.15, Momentum Constant = 0.9

A \ P	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	78.9	-	1.67	2.76	14.3	2.37
Bus	-	81.4	1.63	7.6	1.14	8.23
Car	0.8	3.2	83.4	10	2.1	0.5
Mini Truck/van	-	3.5	9.8	79.33	4.8	2.57
Auto Rickshaw	4.65	2.8	8.7	11.3	71.6	0.95
Truck	-	18.5	6.4	2.8	2.0	70.3

6. Neural Network with 2 Hidden Layer
   Learning rate = 0.25, Momentum Constant = 0.9

A \ P	Motor Bike	Bus	Car	Mini Truck / Van	Auto Rickshaw	Truck
Motor Bike	83.5	-	2.6	6.8	6.8	0.3
Bus	-	86.3	1.7	5.8	0.1	6.1
Car	0.2	6.2	81.08	11.4	1.1	0.02
Mini Truck/van	-	2.8	10.3	84.76	2.04	0.01
Auto Rickshaw	5.8	0.76	8.17	2.3	82.1	0.87
Truck	-	12.1	3.4	0.3	0.45	83.75

4.2.4 Observation and Analysis

The proposed approach works well for vehicle detection from video. But there are still some noise present in while segmenting the vehicles from video. Classification of vehicles using neural network works best for learning rate = 0.25 and momentum constant = 0.9. Here parameters are tuned in experimental way.

Chapter 5

Conclusion and Future Work

It can be concluded from the obtained results that, vehicles are segmented successfully from the static background. Also results of implemented techniques for detection and classification is observed. From this observations the attempts for the improvement in the results will be made in future. Classification of vehicles are performed in experimental way. It can improved in future, by using the optimize parameter tuning function. The next step to be performed is test the system in real time detection and classification and also recognize the vehicles.

References

[1] B. B. K. Toyama, J. Krumm and B. Meyers, "Wallower: principles and practice of background maintenance," *7th IEEE International Conference on Computer Vision*, vol. 1, pp. 255–261, 1999.

[2] D. S. T. C. L. Cheng, S.Wang and S. V. N. Vishwanathan, "An online discriminative approach to background subtraction," *IEEE International Conference on Advanced Video and Signal based Surveillance*, 2006.

[3] M. Cristani and V. Murino, "Background subtraction with adaptive spatiotemporal neighborhood analysis," *3rd International Conference on Computer Vision Theory and Applications*, pp. 484–489, 2008.

[4] S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Visual Communications and Image Processing*, vol. 5308, pp. 881–892, 2004.

[5] D. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pp. 192–199, 2008.

[6] A. P. S. Calderara, R. Melli and R. Cucchiara, "Reliable background suppression for complex scenes," *4th ACM international workshop on Video surveillance and sensor networks*, pp. 211–214, 2006.

[7] D. H. I. Haritaoglu and L. Davis, "W4: A real time system for detecting and tracking people," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998.

[8] M. P. R. Cucchiara, C. Grana and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[9] B. R. N. M. Oliver and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 831–843, 2000.

[10] W. H. X. Luo, S. M. Bhandarkar and H. Gu, "Nonparametric background modeling using the condensation algorithm," *IEEE International Conference on Video and Signal Based Surveillance*, pp. 1–6, 2006.

[11] D. H. A. M. Elgammal and L. S. Davis, "Non-parametric model for background subtraction," *6th European Conference on Computer Vision*, pp. 751–767, 2000.

[12] N. J. B. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine Vision and Applications*, pp. 187–193, 1990.

[13] T. D. C. R. Wren, A. Azarbayejani and A. Pentland, "Pfinder: Realtime tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[14] R. R. W. E. L. Grimson, C. Stauffer and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 22, 1998.

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

[16] W. Kim and C. Kim, "Background subtraction for dynamic texture scenes using fuzzy color histograms," *Signal Processing Letters, IEEE 19.3*, pp. 127–130, 2012.

[17] D. H. L. D. K. Kim, T. H. Chalidabhongse, "Background modeling and subtraction by codebook construction," *IEEE International Conference on Image Processing, ICIP*, 2004.

[18] V. B. D. Butler, S. Sridharan, "Real-time adaptive background segmentation," *International Conference on Acoustics, Speech, and Signal Processing, ICASSP, Hong Kong, China*, 2003.

[19] K. I. and M. S.S., "Frame differencing with post-processing techniques for moving object detection in outdoor environment," *IEEE Conference Publication*, 2011.

[20] L. C. J.B. Pineau, "Gaussian mixture classification for moving object detection in video surveillance environment," *Image Processing, 2005. ICIP 2005. IEEE International Conference*, vol. 3, pp. 113–16, September 2005.

[21] N. L. H. W. X. Zheng, Y. Zhao, "An automatic moving object detection algorithm for video surveillance application," *Embedded Software and Systems, ICESS '09. International Conference*, pp. 541–543, May 2009.

[22] J. P. M. Sagrebin, "Real time moving object detection for video surveillance," *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 31–36, September 2009.

[23] Z. B. Y. Liu, "The improved moving object detection and shadow removing algorithm for video surveillance," *Computational Intelligence and Software Engineering (CiSE), International Conference*, pp. 1–5, December 2010.

[24] P. S. R. T. S. Kumar K., S. Prasad, "Multiple cameras using real time object tracking for surveillance and security systems," *Emerging Trends in Engineering and Technology (ICETET), 3rd International Conference*, pp. 213–218, November 2010.

[25] D. B. M.T.G. Krishna, M. Ravishankar, "Automatic detection and tracking of moving object in complex environment for video surveillance application," *Electronics Computer Technology (ICECT), 3rd International Conference*, vol. 1, pp. 234–238, April 2011.

[26] L. W. Hu, T. Tan and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 334–352, 2004.

[27] K. C. Antony, "Vehicle detection based on feature extraction and svm classification," *IOSR Journal of Computer Engineering*, 2014.

[28] R. F. K. M. N. P. P. S. Gupte, O. Masoud, "Detection and classification of vehicles," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 3, March,2002.

[29] Y. C. A. Khammari, F. Nashashibi, "Vehicle detection combining gradient analysis and adaboost classification," *IEEE Conference on Intelligent Transportation Systems*, September,2005.

[30] M. Sivaraman, S.Trivedi, "A general active learning framework for on-road vehicle recognition and tracking," *IEEE Transaction on Intelllgent Transportation Systems*, 2010.

[31] S. T. L. T. Ng, S. A. Suandi, "Vehicle classification using visual background extractor and multi-class support vector machines," *The 8th International,Conference on Robotic, Vision,Signal Processing and Power Applications*, 2014.

[32] M. Sivaraman, S.Trivedi, "Active learning for on-road vehicle detection: acomparative study," *Machine Vision and Applications,Springer*, November,2011.

[33] M. L.Abraham, "Vehicle detection and classification from high resolution satellite images," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-1, 2014.

[34] "Dataset videos." https://drive.google.com/open?id=0B9xnfMdqeLz0fmNtQUdBZG1aN0V3TmJxSTc1V0dYb3otQS03WExmeW5qZVZNamdQV2U1VmM&authuser=0.

[35] D. G. Lowe, "Object recognition from local scale space-invariant features," *International Journal of computer vision*, vol. 60, pp. 91–110, January 2004.

[36] "Neural network." http://www.fon.hum.uva.nl/praat/manual/Feedforward_neural_networks.html.

[37] R. V. Vu N.P. Dao, "A performance comparison of different back propagation neural networks methods in computer network intrusion detection,"

[38] http://in.mathworks.com/.