Automatic Vehicle Detection, Tracking and Recognition of License Plate in Real Time Videos

Lucky Kodwani

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769008, Odisha, India

Automatic Vehicle Detection, Tracking and Recognition of License Plate in Real Time Videos

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Technology

in

Communication and Signal Processing

by

Lucky Kodwani
(Roll: 211EC4105)

Under the guidance of

Prof. Sukadev Meher

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Department of Electronics and Communication Engg
National Institute of Technology Rourkela
ROURKELA-769 008, ODISHA, INDIA

May 29, 2013

Certificate

This is to certify that the thesis titled as **Automatic Vehicle Detection, Tracking and Recognition of License Plate in Real Time Videos** by **Lucky Kodwani** is a record of an original research work carried out under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of Master of Technology degree in **Electronics and Communication Engineering** with specialization in **Communication and Signal Processing** during the session 2012-2013.

Sukadev Meher

Professor

Acknowledgement

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis. Foremost, I would like to express my sincere gratitude to my advisor, Prof. Sukadev Meher for providing me with a platform to work on challenging areas of vehicle detection, tracking and automatic license plate recognition for real time videos. His profound insights and attention to details have been true inspirations to my research.

Next, I would like to express my respects to Prof. S. K. Patra, Prof. K.K.Mahapatra , Prof. S. K. Behera , Prof. Poonam Singh, and Prof. A.K. Sahoo for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all the faculty members and staff of the Department of electronics and communication engineering, N.I.T.Rourkela for their generous help in various ways for completion of this thesis.

I would like to thank my lab mates Mr Deepak Panda, Mr Adiya Acharya, Mr Deepak Singh and Mr Bodhisattwa Charkraborty for their time to time advice and encouragement in my project work. Their help can never be penned with words. I would like to thank all my friends and especially my classmates Sonia Das for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious and Imran Khan for his help and support.

Most importantly, none of this would have been possible without the love and patience of my family. My family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heartfelt gratitude to them especially to my mother.

Lucky Kodwani

Abstract

Automatic video analysis from traffic surveillance cameras is a fast-emerging field based on computer vision techniques. It is a key technology to public safety, intelligent transport system (ITS) and for efficient management of traffic. In recent years, there has been an increased scope for automatic analysis of traffic activity. We define video analytics as computer-vision-based surveillance algorithms and systems to extract contextual information from video. In traffic scenarios several monitoring objectives can be supported by the application of computer vision and pattern recognition techniques, including the detection of traffic violations (e.g., illegal turns and one-way streets) and the identification of road users (e.g., vehicles, motorbikes, and pedestrians). Currently most reliable approach is through the recognition of number plates, i.e., automatic number plate recognition (ANPR), which is also known as automatic license plate recognition (ALPR), or radio frequency transponders.

Here full-featured automatic system for vehicle detection, tracking and license plate recognition is presented. This system has many applications in pattern recognition and machine vision and they ranges from complex security systems to common areas and from parking admission to urban traffic control. This system has complex characteristics due to diverse effects as fog, rain, shadows, uneven illumination conditions, occlusion, variable distances, velocity of car, scene's angle in frame, rotation of plate, number of vehicles in the scene and others. The main objective of this work is to show a system that solves the practical problem of car identification for real scenes. All steps of the process, from video acquisition to optical character recognition are considered to achieve an automatic identification of plates.

Detection of moving objects in video streams is the first relevant step of information and background subtraction is a very popular approach for foreground segmentation. In this thesis, different background subtraction methods are simulated to overcome the problem of illumination variation, shadows, background clutter and camouflage. Next step is License plate extraction which is an important stage in license plate recognition for automated transport system. We are proposing two methods for extraction of license plates and comparing it with other existing methods. The Extracted license plates are segmented into individual characters by using a region-based method. The recognition scheme combines adaptive iterative thresholding with a template matching algorithm. The

method is robust to illumination, character size and thickness, skew and small character breaks. The main advantages of this system is its real-time capability and that it does not require any additional sensor input (e.g. from infrared sensors) except a video stream. This system is evaluated on a large number of vehicle images and videos. The system is also computationally very efficient and it is suitable for others related image recognition applications. This system has wide range of applications such as access control, tolling, border patrol, traffic control, finding stolen cars, etc. Furthermore, this technology does not need any installation on cars, such as transmitter or responder.

Contents

List of Figures

CHATER-4

List of Tables

Chapter -2

Chapter-3

Chapter -4

Chapter 1

Introduction

The escalating increase of contemporary urban and national road networks over the last three decades emerged the need of efficient monitoring and management of road traffic. Conventional techniques for traffic measurements, such as inductive loops, sensors or EM microwave detectors, suffer from serious shortcomings, expensive to install, they demand traffic disruption during installation or maintenance, they are bulky and they are unable to detect slow or temporary stop vehicles. On the contrary, systems that are based on video are easy to install, use the existing infrastructure of traffic surveillance. Furthermore, they can be easily upgraded and they offer the flexibility to redesign the system and its functionality by simply changing the system algorithms. Those systems allow measurement of vehicle's speed, counting the number of vehicles, classification of vehicles, and the identification of traffic incidents (such as accidents or heavy congestion).There is a wide variety of systems based on video and image processing employing different methodologies to detect vehicles and objects.

1.1 Traffic surveillance:

Traffic surveillance system is an active research topic in computer vision that tries to detect, recognize and track vehicles over a sequence of images and it also makes an attempt to understand and describe object behaviour, vehicle activity by replacing the aging old traditional method of monitoring cameras by human operators. A computer vision system can monitor both immediate unauthorized behaviour and long term suspicious behaviour, and hence alerts the human operator for deeper investigation of the event. The video surveillance system can be manual, semi-automatic, or fully-automatic depending on the human involvement. Human operator is responsible for monitoring in manual video surveillance system. The entire task is done by him by watching the visual information coming from the different cameras. It's a tedious and arduous job of an operator to watch the multiple screens and at the same time to be vigilant from any unfortunate event. These systems are proving to be ineffective for busy large places as the number of cameras exceeds the capability of human experts. Such systems are in widespread across the world. The semi-automatic traffic surveillance system takes the help of both human operator and computer vision. The object is being tracked by the computer vision algorithm and the job of classification, personal identification, and activity recognition is done by the human operator. Lower level of video processing is used in these systems, and much of the task is done with the help of human operator intervention. In the fully-automatic system there is no human intervention and the entire job is being done by the computer vision. These systems are intelligent enough to automatically track, classify, and recognise the object. In addition, it intelligently detects the suspicious behaviour and does the activity recognition of the object.

In urban environment, monitoring congestion across the road, vehicle interaction and detection of traffic rule violation can be done with visual surveillance systems [1]. The Video surveillance system can prevent serious accidents, so that precious lives can be saved. ANPR is a very specialized well-researched application for video analytics. Toll stations of freeways have dedicated lanes with cameras, where registered users can slowly pass without stopping [2]. In contrast, inner city congestion charge systems (e.g., Stockholm, Sweden; London, U.K., and Singapore) have to be less intrusive and operate on the normal flow of passing traffic (free-flow tolling).

For most traffic surveillance systems there are three major stages which are used for estimation of desired traffic parameters i.e. vehicle detection, tracking, and classification. For detection of vehicles, most of the methods [6]-[12] assume that the camera is static and then desired vehicles can be detected by image differencing. Tracking is a very important issue in computer vision. Recently, there is a deep interest in surveillance applications. The main purpose of tracking in computer vision is to recognize and locate a prototype in a series of sequential frames. Many applications are based on tracking such as video processing, security, surveillance and automatic procedures. Then, different tracking scheme are designed to track each vehicle. After that, several vehicle features like shape, length, width, texture, license number etc., are extracted for vehicle classification. Every visual surveillance systems start with detecting moving object in video streams.[19]

Traffic surveillance system can provide effective and efficient application ranging from commercial and public security, Military security, visual surveillance, crowd flux statistics and congestion analysis, person identification, detection of anomalous behaviour, etc.

1.2 Overview of the proposed model

A typical surveillance system consists of a traffic camera network, which processes captured traffic video on-site and transmits the extracted parameters in real time. Here our focus is on the study of algorithmic part of such a system.

In this thesis, we present full-featured vehicle detection, tracking and license plate recognition system framework, particularly designed to work on video footage. This system mainly having four modules:-

- Video Acquisition
- Vehicle detection and tracking
- License plate extraction
- Character recognition unit

This system is used to detect, recognize and track vehicle from incoming video frames in dynamic scenes then extract the license plate from it as shown in Figure 1.1. It has found numerous applications as wide as possible such as: access control in security sensitive areas, securities for communities and important buildings, detection of military target areas, traffic surveillance in cities and highways, detection of anomalies behaviour, traffic control management for recognize vehicles that commit traffic violation, such as occupying lanes reserved for public transport, breaking speed limits, crossing red light, entering restricted area without permission; and among many other applications.

The system is designed for real time videos where a camera is used for continuous recording of videos. The view of camera or the area covered by camera is fixed between entry zone and exist zone. Each frame is continuously processed to check the presence of a vehicle. A defined connected component area is taken as threshold; if the detected area

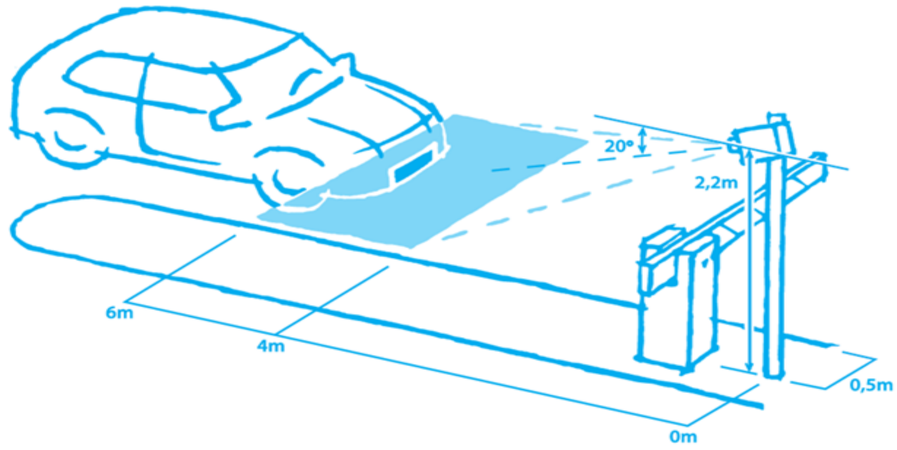


Figure 1. 1 ANPR system implementation in practical scenario

is above that threshold value then it will be recognized as a vehicle and will be tracked. A distance is defined between the vehicle and the camera and when the vehicle comes within that range *i.e* vehicle's connected component area is maximum, these frames of video are passed to license plate recognition algorithm. After that recognition of character takes place and data is stored and compare with data base. The flow chart of the system is shown in Figure 1.2.

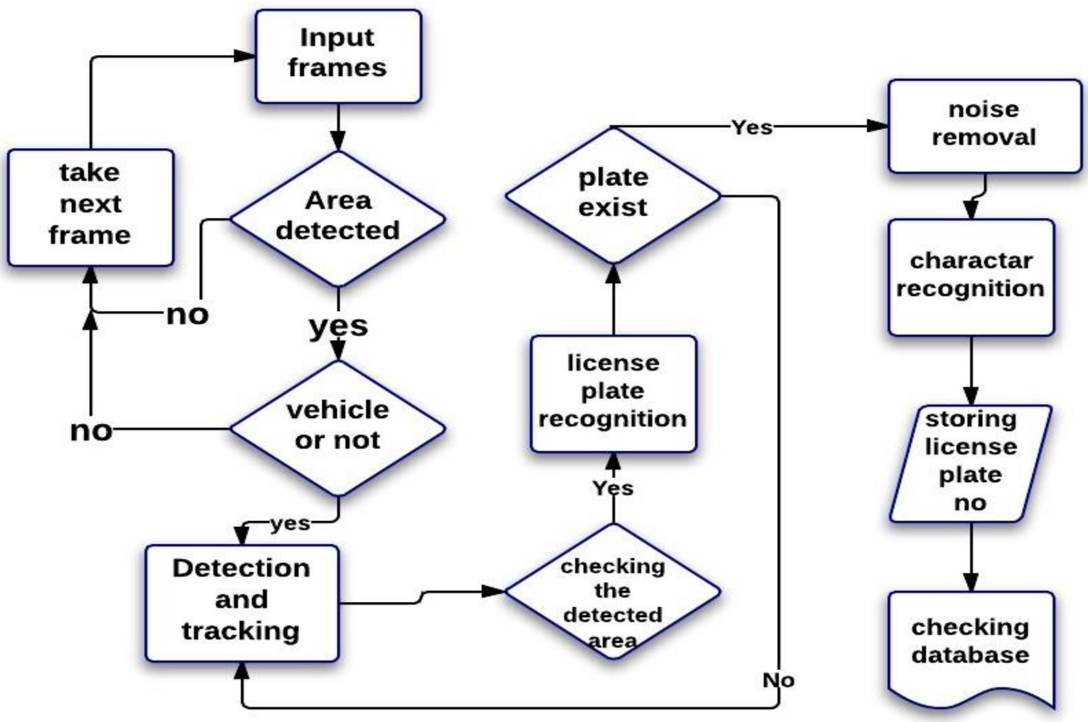


Figure 1. 2 Flow chart of the system

.

1.2.2 Block Diagram of the Proposed System

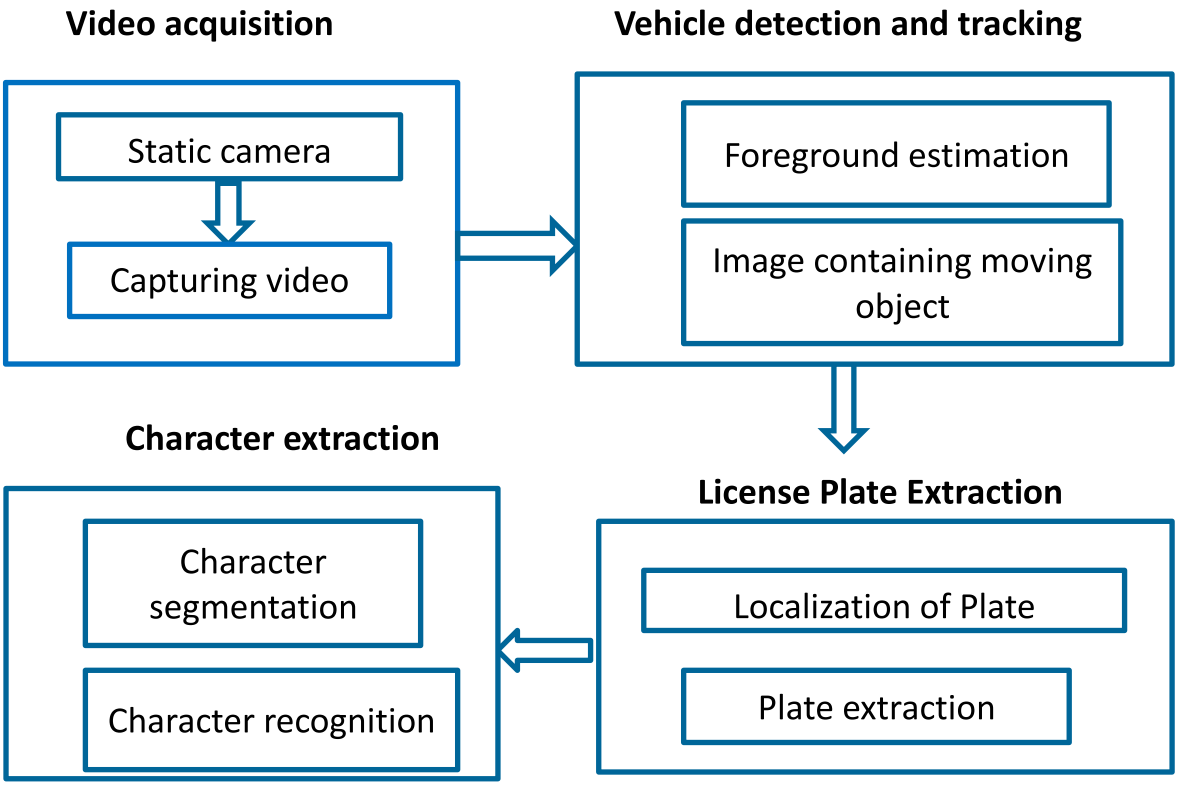


Figure 1. 3 Block diagram of the system

This Proposed System having mainly four modules:

- *<u>Video Acquisition</u>* – In this module videos are taken by the static camera situated at traffic scenario. A camera network that has the ability to transmit images in real time to a central operational centre. The processing of the images can be carried out on-site saving valuable network bandwidth as it transmits only the outcome of the calculations. The whole process can also be performed either in real time video streaming from an operational centre or in already stored video material.

- *<u>Vehicle Detection And Tracking</u>* – In vehicle detection we have simulated various background subtraction techniques available in the literature. The background subtraction technique should overcome the problems of varying illumination condition, background clutter, camouflage and shadow. Motion segmentation of foreground object has been done in real time. It's hard to get this entire problem solved in one background subtraction technique. So the idea was to simulate and evaluate their performance on various video data taken in different situations.

- *License Plate Extraction* - License plates are first located in current frame then they are extracted using various available techniques in the literature based on Hough Transform method, Template matching technique, Region growing algorithm, Histogram Approach and Edge Detection Approach.

- Character Extraction – Images of the extracted plates are the input to this module. Here first license plate image is cropped in lines, and then characters are segmented and recognized.

1.2 Motivation

Traffic surveillance is the most active research topic in computer vision for humans and vehicles. Our aim is to develop an intelligent automatic License plate recognition in real time surveillance videos and replacing the age old traditional method of monitoring by human operators. Now a day the license plate recognition has been essential due to the rapidly growing number of cars. The increasing number of automobiles has facilitated human life but it has also lead to various issues of traffic congestions, parking problems, accidents etc. Our motivation in doing is to design a traffic surveillance system for vehicle detection, tracking and license plate recognition in real time videos using visual surveillance techniques.

1.3 Problem statement

To design and develop a real-time detection, tracking and license plate recognition system that will work efficiently under the conditions of slow moving objects and the objects that are merged into the background due to a temporary stop and becoming foreground again, adaptive to different traffic environment conditions, robustness against progressive or sudden illumination changes, Occlusions, identification time of the system should be as short as possible. The system should detect all the types of vehicles, recognize all the license plates of the country and should also be resistant to any kinds of disturbances, which may occur in images and mechanical plate damages which may appear in reality.

The attributes of the License plates play important role in the recognition process. The size, colour of the license plate, its font face i.e. size, colour of each character, spacing between characters, the number of lines in the license plate, script, characters' height and width are maintained very strictly in developed countries. There are several countries in the world who have adapted this very method of standardizing the License plate. Some of the standard license plates used in developed countries are shown in Figure 1.4.

Figure 1. 4 standard license plates used in developed countries

But in India, the license plate is purely localized and people don't follow the standard pattern assigned by Indian government, so the recognition process is quite difficult. Some of the Indian license plates with variations in shape, script, etc. are shown in Figure: 1.5.

Figure 1. 5 Indian license plates

Apart from these conditions, the algorithms applied for the recognition also plays a vital role. If the quality of the algorithm is good, then more varieties of images can be given as input to the system, and this will also reduce the computation speed of the process. The most basic issues in the real-time ALPR system is the accuracy and the recognition speed.

Every system should fulfil the following demands:

- The identification time of the system should be as short as possible.
- The system should detect all the types of vehicles.
- The system should recognize all the license plates of the country.
- The system should also be unaffected to any kinds of disturbances, which may occur in images and mechanical plate damages which may appear in reality.

Other than this, there are several other parameters on which the quality of the recognition depends. These limitations are

- High quality imaging – the quality of the captured image is high, so that it will be easy to evaluate the attributes of the image.
- Minimal skewing and rotation – the camera is fit such that the captured image or video will not suffer much decent angle of skewing and rotation.
- Better Flash – as the License plate is retro-reflective in nature, so if high illumination will be there, then the LP region will be much more brightened and the attributes of the license plate will not be much clearer.

1.4 Factors influencing the performance of system

Factors, which may have a negative influence on the results of this system, can be classified into a few groups: weather conditions, lighting, license plate placement in the picture, vehicle movement, mechanical plate damages, and other captions in the picture etc. some examples are shown in figure 1.6. Most of the above mentioned factors could be eliminated by the use of proper lighting, specialist equipment recording the videos, designing a workplace and by the use of proper computer image processing techniques. Without such care the system may become a highly complicated problem.

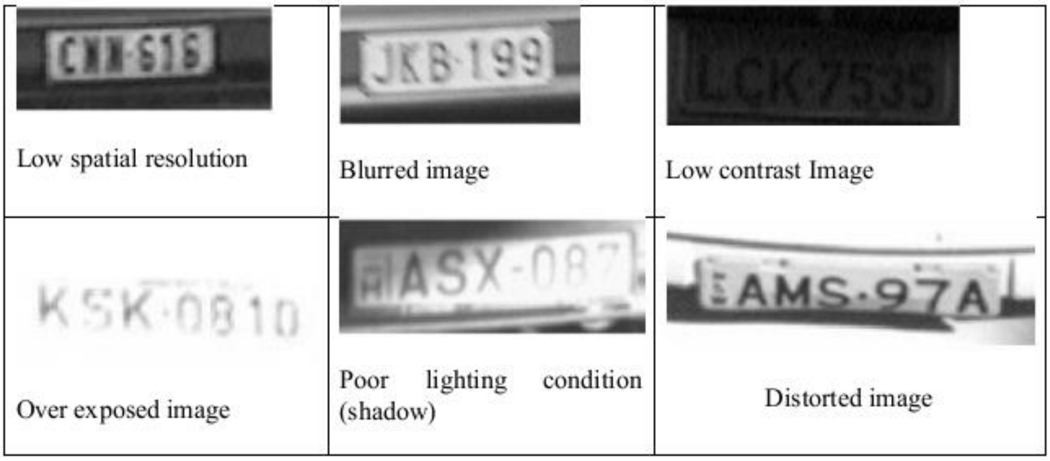


Figure 1. 6 Example of Some Distorted Image

1.5 Organization of the Thesis

The remaining part of the thesis is organized as follows

Chapter 2 presents a brief survey of vehicle detection in real time. Different methods are discussed in detail and their comparisons are shown.

Chapter 3 describes the license plate extraction using different existing methods and proposing two methods and showing their comparisons with other existing methods.

Chapter 4 character segmentation part is discussed.

Chapter 5 concludes the thesis with the some suggestions for future research work.

Chapter 2

Vehicle Detection

Traffic surveillance is used by private companies, governments and public organizations for efficient management of transport networks, road safety, public safety in highways and busy streets. A static camera observing a scene is a common case of a surveillance system. Identifying intruding objects is an important step in analysing the scene and successful segmentation of moving foreground object from the background ensures object classification, vehicle identification, tracking, and activity analysis, making these later steps more efficient.

The task of moving object segmentation is to extract meaningful information about the moving vehicle from video sequences, which provides the convenience of object-based representation and manipulation of video content. It is an essential step for many computer vision applications such as video compression, video retrieval, video surveillance, and pattern recognition. Conventional approaches to moving object segmentation include frame difference methods, background subtraction methods, and optical flow methods [3].

Foreground estimation and segmentation is the first stage of several traffic surveillance systems. The foreground regions are marked (e.g., mask image) for processing in the subsequent steps. There are two main different approaches to estimate the foreground, which both use strong assumptions to comply with the aforementioned definition. First, a background model of some kind can be used to accumulate information about the scene background of a video sequence. The model is then compared to the current frame to identify differences (or “motion”), provided that the camera is stationary. This concept lends itself well for computer implementation but leads to problems with slow-moving traffic. Any car should be considered foreground, but stationary objects are missed due to the lack of motion.[18]

Hu et al. [1] categorized motion detection into three major classes of method as frame differencing, optical flow, and background subtraction.

2.1 Categorization of motion detection

2.1.1 Frame differencing

Frame differencing [4] is a pixel-wise differencing between two or three consecutive frames in an image sequence to detect regions corresponding to moving object such as human and vehicles. The threshold function determines change and it depends on the speed of object motion. If the speed of the object changes significantly, then it’s difficult to maintain the quality of segmentation. The inter-frame differencing approach detects parts of moving objects by comparing two successive frames. But, it can identify only differences in the background and, for that reason; it detects only parts of a vehicle covering the background in the previous frame. Despite some enhancing techniques [4] this approach cannot satisfactorily deal with realistic traffic circumstances where vehicles might remain still for a long time.

2.1.2 Optical Flow

To detect moving regions in an image, optical flow [5] uses flow vectors of the moving objects over time. It is used for motion-based segmentation and tracking applications. It is a dense field of displacement vectors which defines the translation of each pixel region. Optical flow is best suited in presence of camera motion, but however most flow computation methods are computationally complex and sensitive to noise.

2.1.3 Background subtraction

The background subtraction [6], [7], [8], [9], [10], and [11] is the most popular and common approach for motion detection. In this method the current image is subtracted from a reference background image, which is upgraded during a period of time. It works well only in the presence of stationary cameras. The subtraction leaves only non-stationary or new objects, which include whole silhouette region of an object. This approach is simple and computationally affordable for real-time systems, but is extremely sensitive to dynamic scene changes from lightning and extraneous event etc. Therefore it is highly dependent on a good background maintenance model.

Here in this chapter we have simulated different background subtraction techniques available in the literature for motion segmentation of object. The problem with background subtraction [8], [9] is to automatically update the background from the incoming video frame and it should be able to overcome the following problems:

- Motion in the background: Non-stationary background regions, such as leaves and branches of trees, a flag waving in the wind, or flowing water, should be identified as part of the background.
- Illumination changes: The background model should be able to adapt, to gradual changes in illumination over a period of time,
- Memory: The background module should not use much resource, in terms of computing power and memory.
- Shadows: Shadows cast by moving object should be identified as part of the background and not foreground.
- Camouflage: Moving object should be detected even if pixel characteristics are similar to those of the background.
- Bootstrapping: The background model must be maintained even in the absence of training background (absence of foreground object).

2.2 Related Work

A large literature exists concerning moving object detection in video streams and to construct reliable background from incoming video frames. It's hard to get all the above problem solved in one background subtraction technique. So the idea was to simulate different background subtraction techniques available in the literature and evaluate their performance on various video data taken in complex situation.

2.2.1 Simple Background Subtraction

In simple background subtraction technique an absolute difference is taken between every current image $I_t(x,y)$ and the reference background image $B(x,y)$ to find out the motion detection mask $D(x,y)$. The reference image of the background is generally the first frame of a video, without containing foreground object.

$$D(x,y) = \begin{cases} 1, & if|I_t(x,y) - B(x,y)| \geq \tau \\ 0, & otherwise \end{cases} \quad (2.\ 1)$$

Where τ is a threshold, which decides whether the pixel is foreground or background. If the absolute difference is greater than or equal to τ, the pixel is classified as foreground; otherwise the pixel is classified as background.

2.2.2 Running Average

Simple background subtraction cannot handle illumination variation and results in noise in the motion detection mask. The problem of noise can be overcome, if the background is made adaptive to temporal changes and updated in every frame.

$$B_t(x,y) = (1-\alpha)B_{t-1}(x,y) + \alpha\, I_t(x,y) \quad (2.\ 2)$$

Where α is a learning rate. The motion detection mask $D(x,y)$ is calculated as follows:

$$D(x,y) = \begin{cases} 1, & if|I_t(x,y) - B(x,y)| \geq \tau \\ 0, & otherwise \end{cases} \quad (2.\ 3)$$

τ - Threshold value

2.2.3 Simple Statistical Difference (SSD)

Simple Statistical Difference method (SSD) computes the mean $\mu_{x,y}$ and the standard deviation $\sigma_{x,y}$ for each pixel (x, y) in the background image containing K images in the time interval $[t_{0,}t_{k-1}]$.

$$\mu_{x,y} = \frac{1}{K}\sum_{k=0}^{K-1} I_k(x,y) \quad (2.\ 4)$$

$$\sigma_{x,y} = \left(\frac{1}{K}\sum_{k=0}^{K-1}(I_k(x,y) - \mu_{x,y})^2\right)^{1/2} \quad (2.\ 5)$$

For motion detection, difference between the current image $I_t(x,y)$ and the mean $\mu_{x,y}$ from the background images is calculated.

$$D(x,y) = \begin{cases} 1, & if\ \left(I_t(x,y) - \mu_{x,y}\right) \geq \lambda\sigma_{x,y} \\ 0, & otherwise \end{cases} \quad (2.6)$$

2.2.4 Motion Detection Based on Sigma – Delta Estimation (∑ -Δ)

At each frame, $\Sigma - \Delta$ (SDE) estimates the background by incrementing the value by one if it is smaller than sample, or decremented by one if it is greater than the sample [11]. This algorithm uses the sign function sgn which is defined as

$$sgn(a) = \begin{cases} -1, & if\ a < 0 \\ 0, & if\ a = 0 \\ 1, & if\ a > 0 \end{cases} \quad (2.7)$$

The background estimated by this method is an approximation of the median of I_t(x, y).The absolute difference between I_t(x, y) and B_t(x, y) gives the difference $\Delta_t(x,y)$. The binary motion detection mask D(x, y) is computed from the comparison of difference $\Delta_t(x,y)$ and time variance $V_t(x,y)$. If $\Delta_t(x,y)$ is smaller than time variance $V_t(x,y)$, it corresponds to background pixel or otherwise it is the foreground pixel. The time variance $V_t(x,y)$ of the pixels signifies their motion activity measure. As described in Table 2.1.

Table2. 1: Basic Sigma-Delta Background Estimation

Algorithm 1: Background detection based on sigma - Delta

Result : Binary motion detection mask $D_t(x, y)$

Initialization

$B_0(x, y) = I_0(x, y)$ // Initialize background model B

$V_0(x, y) = 0$ // Initialize variance V

N = 4

For each frame(t) do

  For each pixel (x, y) do

    $\Delta_t(x, y) = |B_t(x, y) - I_t(x, y)|$ // Compute current difference

    If $\Delta_t(x, y) \neq 0$

      $V_t(x, y) = V_{t-1}(x, y) + sgn(N \times \Delta_t(x, y) - V_{t-1}(x, y))$

    If $\Delta_t(x, y) < V_t(x, y)$ // compute detection mask D

      $D_t(x, y) = 0$

    Else

      $D_t(x, y) = 1$

    If $D_t(x, y) == 0$ // update background

      $B_t(x, y) = B_{t-1}(x, y) + sgn(I_t(x, y) - B_{t-1}(x, y))$

2.2.5 Sigma-Delta Background Estimation with Confidence Measurement

The adaptation of the basic sigma-delta algorithm introduces a confidence measurement that is tied to each pixel and quantifies the trust the current value of that pixel deserves. This enables a mechanism that provides better balance between adaptation to illumination or background changes and prevention against undesirable background-model contamination. The final goal is to preserve the background model from being corrupted with slow-moving vehicles or vehicles that are motionless for a time gap, without compromising real-time implementation. [12]

Three new images are required with respect to the basic sigma-delta algorithm: 1) the frame counter image(I_t^{FC}) ; 2) the detection counter image (I_t^{DC}); and 3) the confidence image(I_t^{CON}).

The variance image in this algorithm represents the variability of pixel intensities when no objects are over that pixel. A low variance should be interpreted as having a "stable background model" that has to be maintained. A high variance should be interpreted as "the algorithm has to look for a stable background model." The main background and variance-selective updating mechanism is linked to the "refresh period." Each time that this period expires, then updating action is taken.

The detection ratio (I_t^{DC}/ I_t^{FC}) can be used as an estimation of the traffic flow. The values of this detection ratio above 80% are typically related to the presence of stopped vehicles or traffic congestion over the corresponding pixels. If this is not the case, then the updating action is permitted.

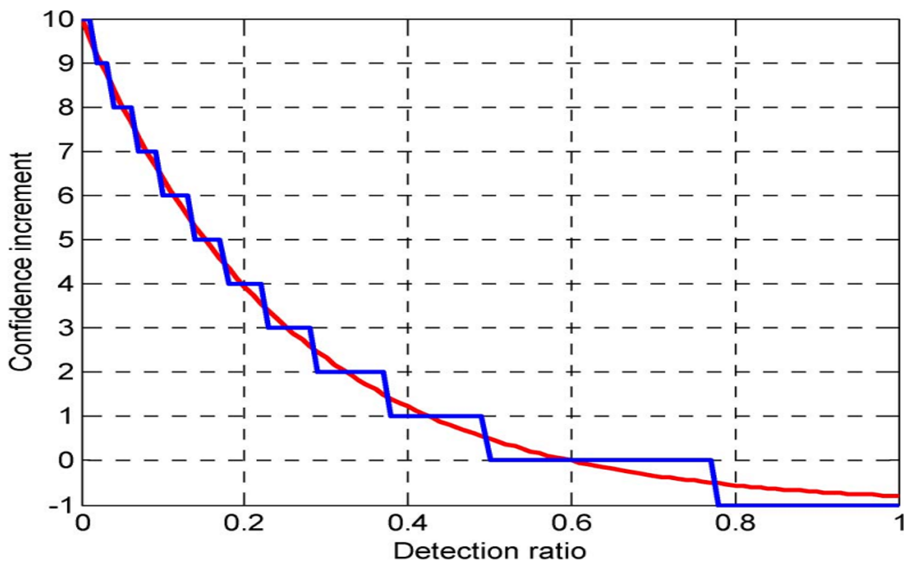


Figure 2. 1 : Function used for incremental update of the confidence measurement.

A parallel mechanism is set up to update the confidence measurement. The principle mechanism is, higher the confidence level is, the lower the updating need for the corresponding pixel is. This is controlled by "confidence period" which is not a constant period of time depends on the confidence itself, for each particular pixel. Specifically, the confidence period length is given by a number of frames equal to the confidence value at the corresponding pixel. Each time the confidence period expires, the confidence measure

Table2. 2: Sigma-Delta with Confidence Measurement

Algorithm 2: Sigma – Delta With Confidence Measurement

Result : Binary motion detection mask $D_t(x,y)$

// Initialize background model and variance

$\boldsymbol{B_0 = I_0; \quad V_0 = v_{ini}}$

// Initialize detection, frame counter and confidence measure images

$\boldsymbol{I_0^{DC} = I_0^{FC} = 0; \quad I_0^{CON} = c_{ini}}$

For each frame t

  $I_t^{FC} = I_t^{FC} + 1;$

  // Confidence Period evaluation and Background updating decision making

  If $I_t^{FC} < I_t^{CON}$

    If I_t^{FC} *is maltiple of* P //setting refresh period

      If $V_t \le v_{th}$

        If $(I_t^{DC} / I_t^{FC}) \le 0.8$

          $U_t = 1$

        End if

      End if

    End if

  Else

    If $V_t \le v_{th}$

      I_t^{CON} **+=** $\gamma\,(I_t^{DC} / I_t^{FC})$

      If $I_t^{CON} == c_{min}$

        $U_t = 1$

      End if

    Else

      $U_t = 1$

    End if

    $I_t^{DC} = I_t^{FC} = 0$

  End if

  // Background updating and detection

    If $U_t == 1$

      $B_t = B_{t-1} + sgn(I_t - B_{t-1})$

      $\Delta_t = |\, B_t - I_t|$

      $V_t = V_{t-1} + sgn(v_{min} + N \times \Delta_t - V_{t-1})$

      $D_t = (\Delta_t \ge V_t)$

    Else

      $\Delta_t = |\, B_t - I_t|$

      $D_t = (\Delta_t \ge V_t)$

    End if

  $\boldsymbol{I_t^{DC} += (D_t == 1)}$

End for

is incrementally updated according to an exponentially decreasing function of the detection ratio as shown in Figure 2.1

$$d: \ \gamma(d) = round\ (\alpha * \exp(-\beta d) - 1)). \qquad (2.\ 8)$$

The gain α is tuned as the confidence maximum increment where as β defining the increment decay rate. In our implementation $\alpha = 11$; therefore, the maximum confidence increment is ten frames, and $\beta = 4$, which adjusts the crossing of the function with − 0.5 around 75% – 80% of the detection rates. When the confidence is decremented down to minimum, background updating is force to occur.

2.2.6 Gaussian mixture model (GMM)

The GMM methodology models each pixel history as a cluster of Gaussian type distributions and uses an on-line approximation to update its parameters. As per this method, the background is found as the expected value of the distribution corresponding to the most populated cluster [13]. The stability of the Gaussian distributions is evaluated to estimate if they are the result of a more stable background process or a short-term foreground process. Each pixel is classified to be a background if the distribution that represents it is stable above a threshold. The model can deal with lighting changes and repetitive clutter. The computational complexity is higher than standard background subtraction methods. This methodology is greatly improved on grounds of performance by considering recursive equations to adaptively update the parameters of the Gaussian model. [14]- [15]

A pixel at time t is modeled as mixture of K Gaussian [10] distributions. The probability of observing the current pixel value is given by

$$P(X_t) = \sum_{i=1}^{k} w_{i,t} * \eta(X_t, \mu_{i,t}, C_{i,j}) \qquad (2.\ 9)$$

Where $w_{i,t}$, , $\mu_{i,t}$ and $C_{i,j}$ are the estimate weight, mean value and covariance matrix of i_{th} Gaussian in the mixture at time t. $\eta(X_t, \mu_{i,t}, C_{i,j})$ is the Gaussian probability density function equation (2)

$$\eta(X_t, \mu_{i,t}, C_{i,j}) \ = \ \frac{1}{(2\pi)^{\frac{n}{2}} |C|^{1/2}} exp^{-\frac{1}{2}(X_t - \mu_t)^T C^{-1} (X_t - \mu_t)} \qquad (2.\ 10)$$

$$C_{k,t} \ = \sigma_k^2 I$$

2.2.7 W^4 Algorithm for Background Estimation

W^4 [10] Algorithm involves locating moving objects in complex road scenes by implementing an advanced background subtraction methodology; this model is a simple and effective method for segmentation of foreground objects. In the training period each pixel uses three values; minimum $m(x\ ,y)$, maximum $n(x\ ,y)$, and the maximum intensity difference of pixels in the consecutive frames $d(x\ ,y)$, for modelling of the background scene.

Let V be an array containing N consecutive images, $V^i(x,y)$ is the intensity of a pixel location (x,y) in the i th image of V. $\sigma(x,y)$ and $\lambda(x,y)$ are the standard deviation and median value of intensities at pixel location (x,y) in all images in V. The initial background for a pixel location (x, y) is obtained as follows:

$$\begin{bmatrix} m(x,y) \\ n(x,y) \\ d(x,y) \end{bmatrix} = \begin{bmatrix} min_z\{V^z(x,y)\} \\ max_z\{V^z(x,y)\} \\ max_z\left\{\left|V^z(x,y) - V^{z-1}(x,y)\right|\right\} \end{bmatrix} \quad (2.\ 11)$$

$$\text{Where } \left|V^z(x,y) - \lambda(x,y)\right| < 2 * \sigma(x,y)$$

The background is changing, cannot remain same for a long period of time, so the initial background needs to be updated. W^4 uses pixel-based update method to cope with illumination variation and physical deposition of object. W^4 uses change map for background updating.The background reconstruction algorithm is a heuristic that provides a periodically updated background and enhances the efficiency of the well-known background subtraction methodology in case of outdoor environment, and it is a key process for a typical background subtraction algorithm, because it supports the weakest part of it, which is the initialization step.

A detection support map (gS) computes the number of times the pixel (x, y) is classified as background pixel.

$$gS_t(x,y) = \begin{cases} gS_{t-1}(x,y) + 1, & if\ pixel\ is\ background; \\ gS_{t-1}(x,y), & if\ pixel\ is\ foreground \end{cases} \quad (2.\ 12)$$

A motion support map (mS) computes the number of times the pixel (x, y) is classified as moving pixel.

$$mS_t(x,y) = \begin{cases} mS_{t-1}(x,y)+1, & if \quad M_t(x,y)=1\,; \\ mS_{t-1}(x,y), & if\ M_t(x,y)=0\,; \end{cases} \qquad (2.13)$$

Where

$$M_t(x,y) = \begin{cases} 1, & if\ (|I_t(x,y) - I_{t+1}(x,y)| > 2*\sigma) \wedge \\ & (|I_{t-1}(x,y) - I_t(x,y)| > 2*\sigma) \\ 0\,, & otherwise \end{cases}$$

The new background model is given by:

$$[m(x,y), n(x,y), d(x,y)] = \begin{cases} \left[m^b(x,y), n^b(x,y), d^b(x,y)\right] \\ \qquad if\ (gS(x,y) > k*N) \\ \left[m^f(x,y), n^f(x,y), d^f(x,y)\right] \\ \qquad if\ (gS(x,y) > k*N\ \wedge mS < r*N) \\ \left[m^b(x,y), n^b(x,y), d^b(x,y)\right] \\ \qquad otherwise\,; \end{cases}$$

(2. 14)

where *k* and *r* is taken to be 0.8 and 0.1 respectively. The current pixel (x, y) is classified into background and foreground by the following equation.

$$B(x,y) == \begin{cases} 0 & background \quad \begin{cases} ((I_t(x,y) - m(x,y) < kd_ì) \\ \vee\ (I_t(x,y) - m(x,y)) < kd_ì) \end{cases} \\ 1 & foreground \qquad otherwise \end{cases}$$

(2. 15)

2.3 Experimental Results

Performance evaluation of different background subtraction techniques have been tested for traffiv.avi video taken from data base. Specification of traffic.avi video is:

Bits per Pixel = 24

Frame Rate = 15

Height = 120

Number of Frames = 120

Video Format = RGB24

Width = 160

- **Simple Background Subtraction (Frame Difference)**

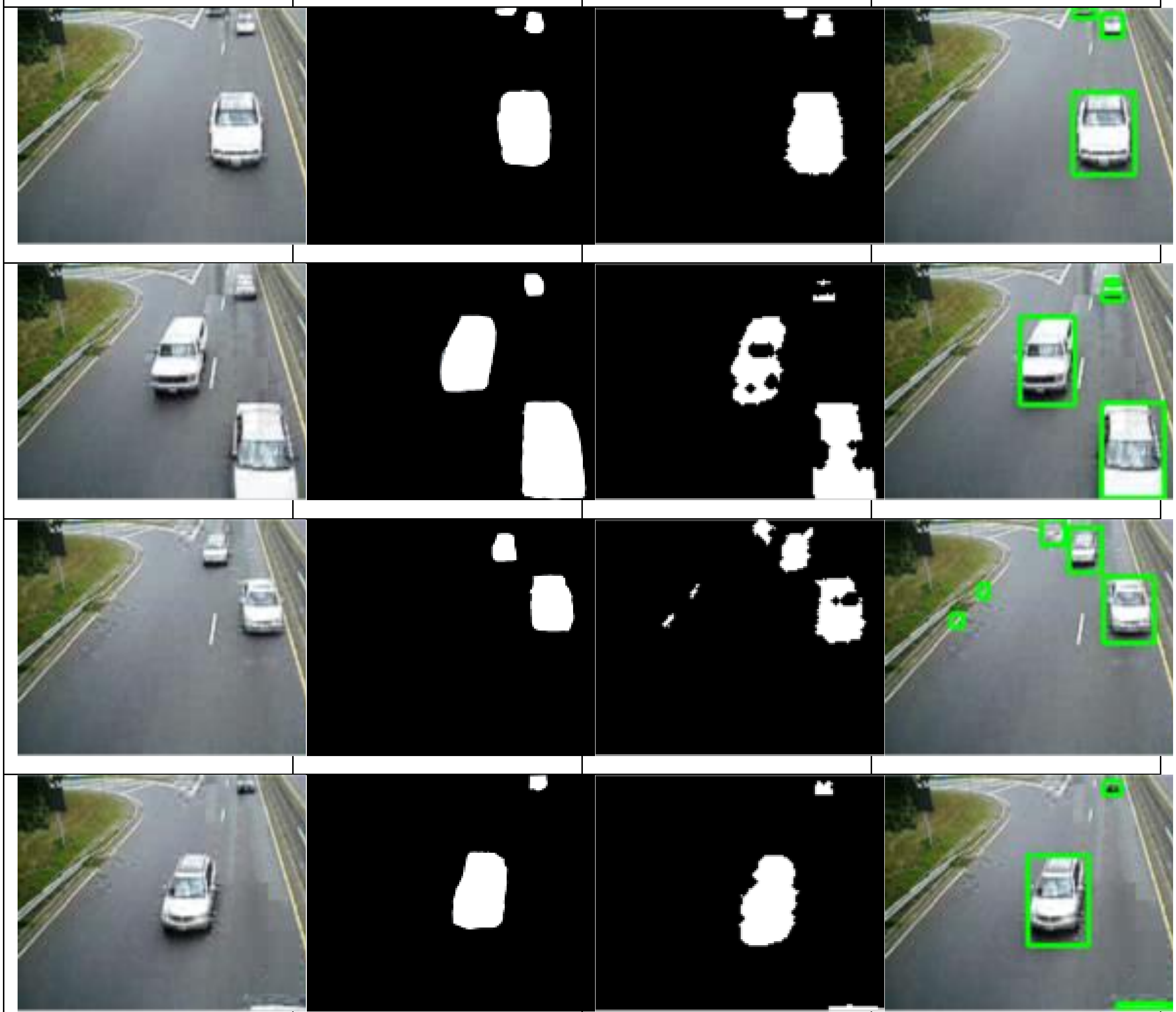
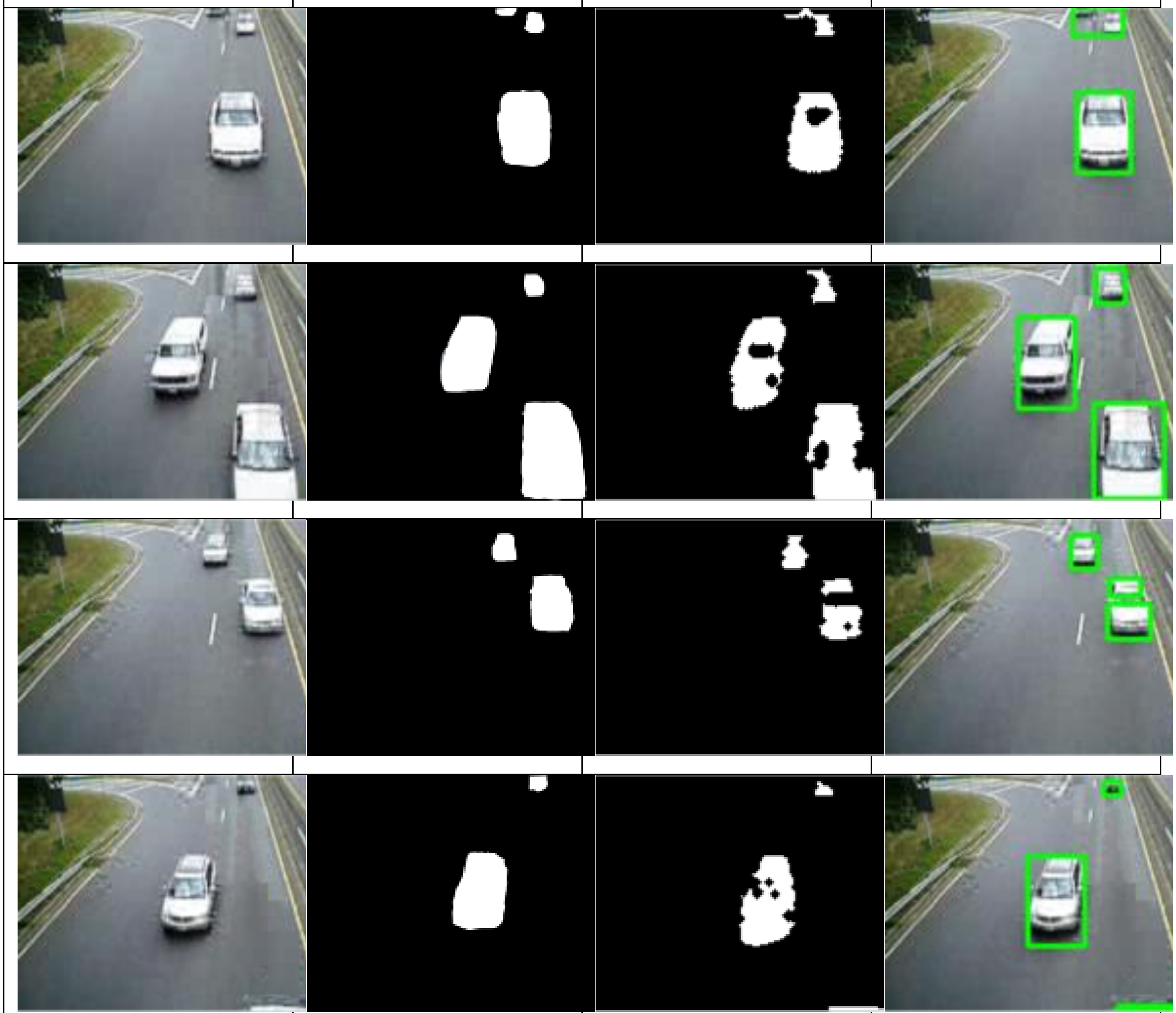
Original video	**Ground truth**	**Foreground detection**	**Tracking**
			
(a)	**(b)**	**(c)**	**(d)**

Figure 2. 2 Results of Frame Difference Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) Frame Diff. Result (d) Tracking

- **Running Average**

Original video	Ground truth	Foreground detection	Tracking



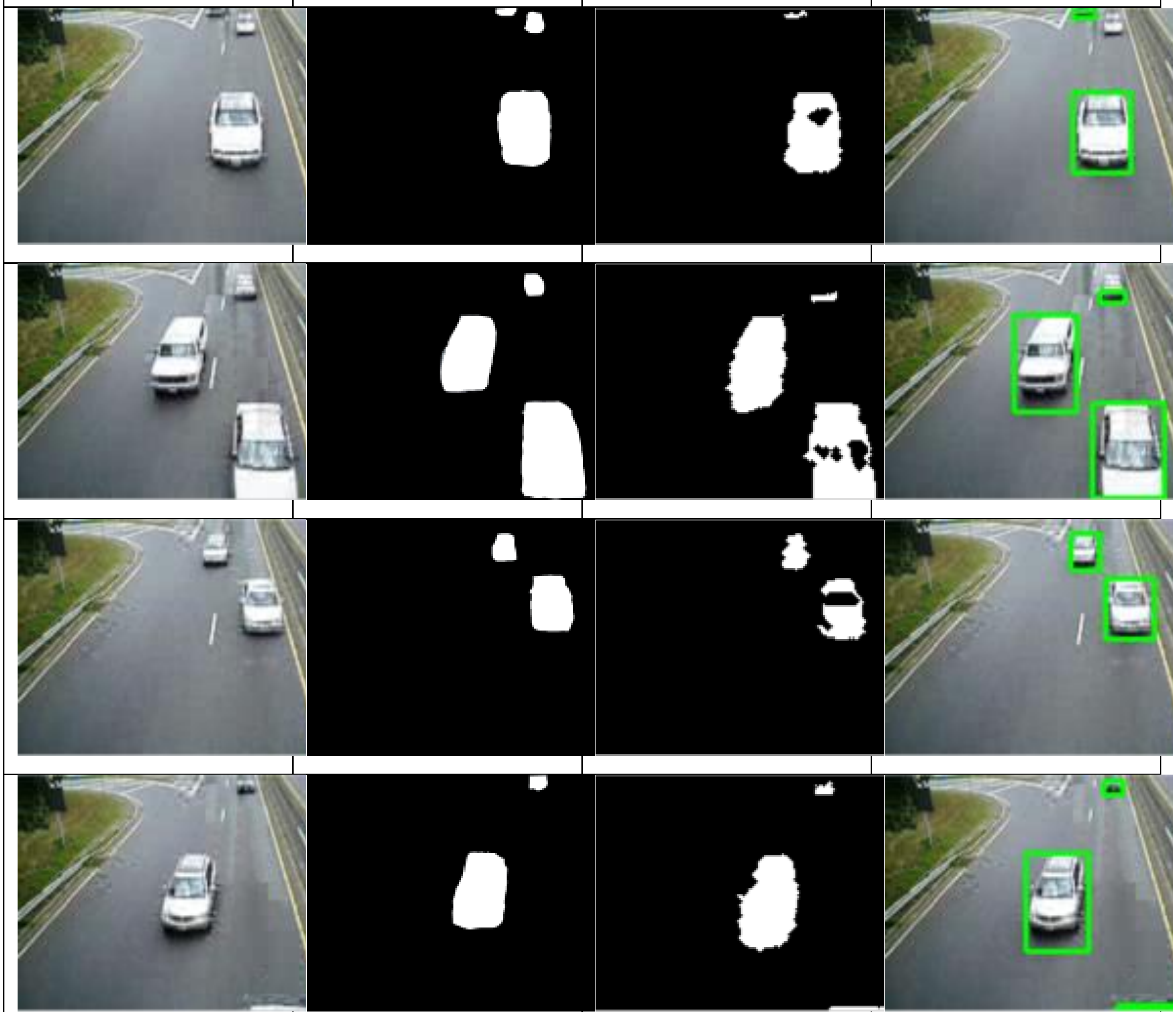
(a)	(b)	(c)	(d)

Figure 2.3 : Results of Running Average Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) Running avg. Result (d) Tracking

- **Simple Statistical Difference (SSD)**

Original video	Ground truth	Foreground detection	Tracking



(a)	(b)	(c)	(d)

Figure 2. 4 : Results of Simple Statistical Difference (SSD) Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) SSD Result (d) Tracking

- **Gaussian mixture model (GMM)**

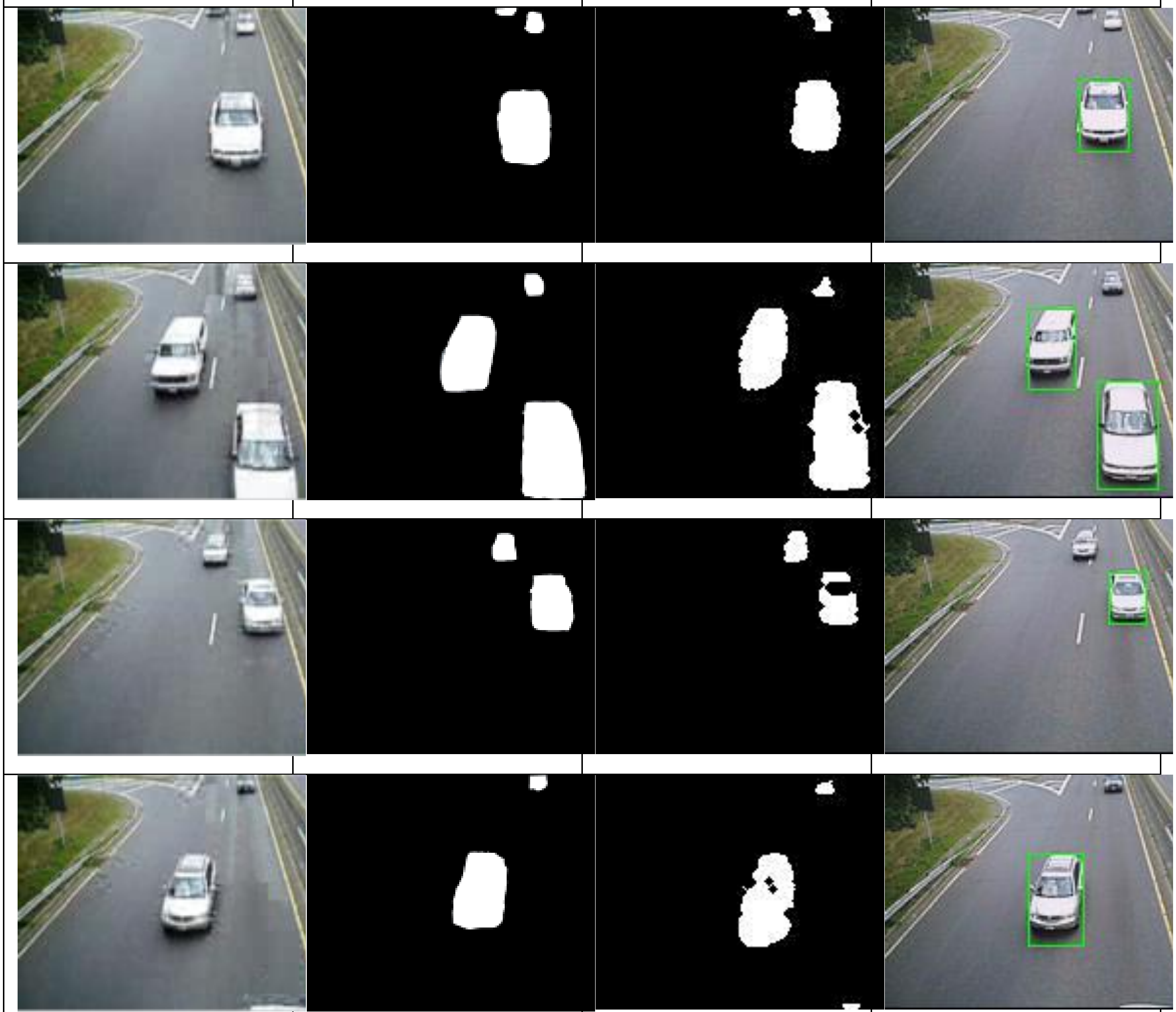
Original video	Ground truth	Foreground detection	Tracking
			
(a)	**(b)**	**(c)**	**(d)**

Figure 2. 5 Results of Gaussian mixture model (GMM) Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) GMM Result (d) Tracking

- **Sigma-Delta Background Estimation with Confidence Measurement**

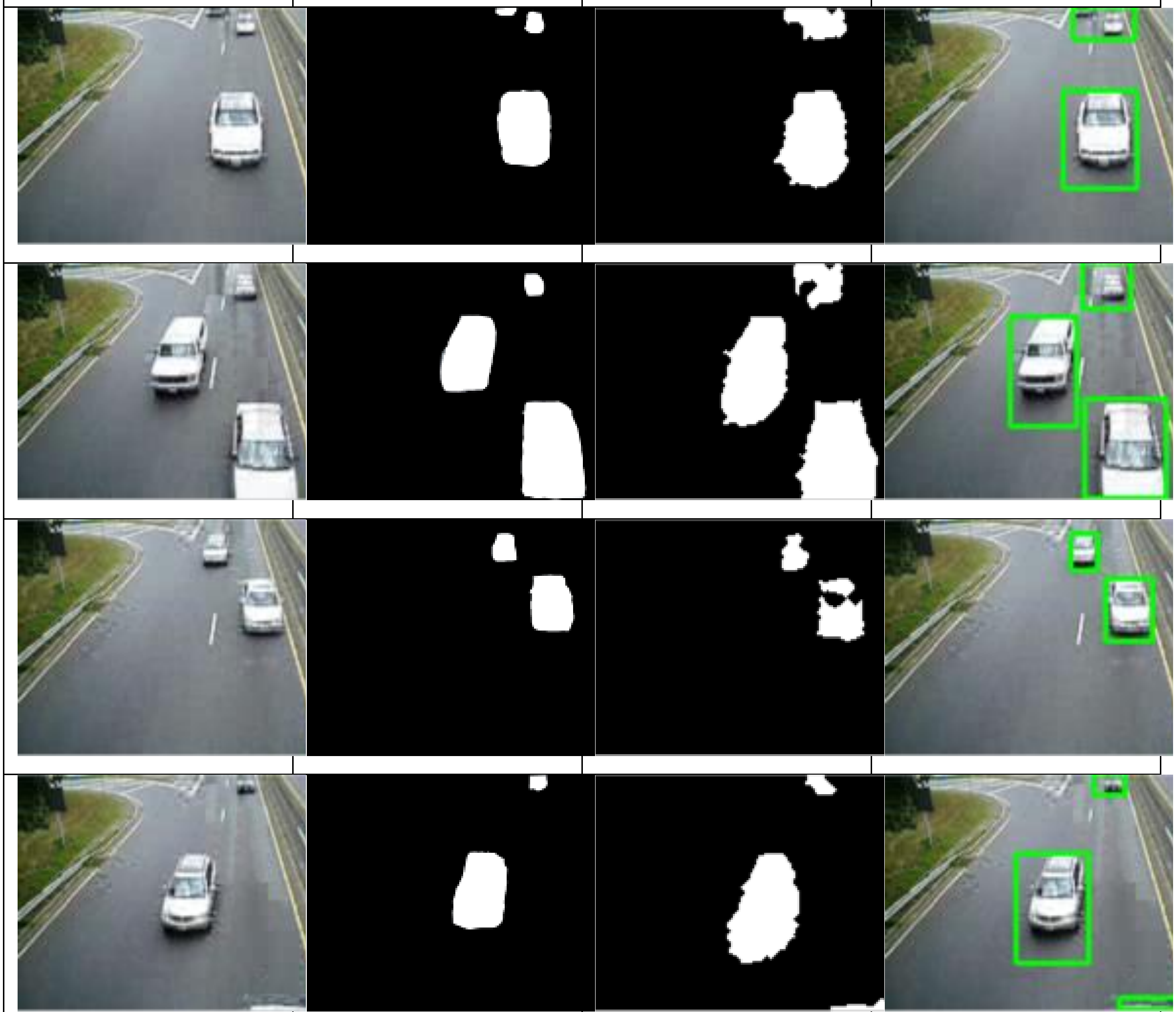
Original video	Ground truth	Foreground detection	Tracking
			
(a)	(b)	(c)	(d)

Figure 2. 6 Results of Sigma-Delta Background Estimation with Confidence Measurement Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) SD with CON Result (d) Tracking

- **W^4 Algorithm for Background Estimation**

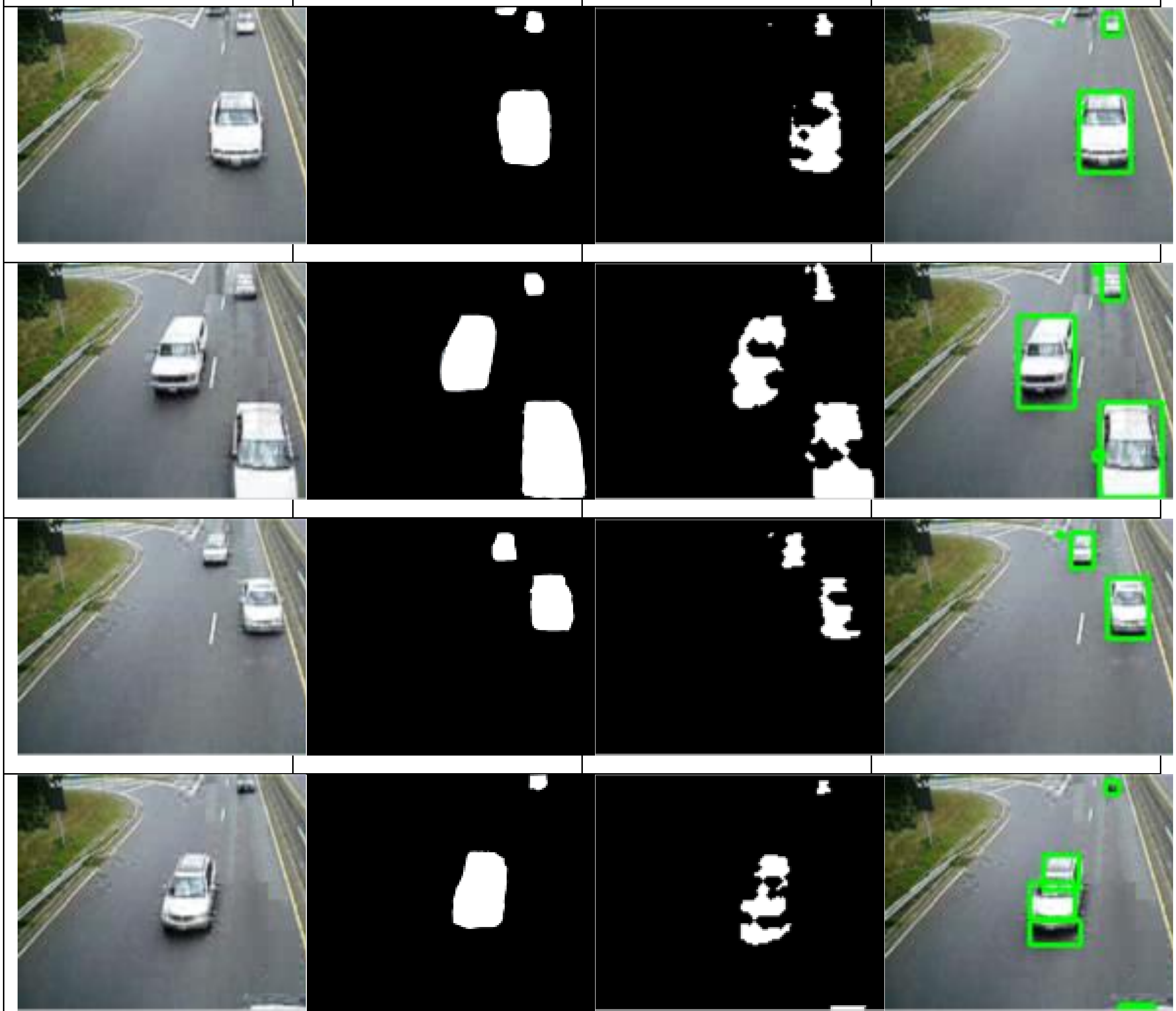
Original video	Ground truth	Foreground detection	Tracking
			
(a)	(b)	(c)	(d)

Figure 2. 7 Results of W^4 Algorithm for Background Estimation Algorithm

(a) Traffic.avi Image (b) Ground Truth (c) W4 Result (d) Tracking

2.3.1 Accuracy Metrics

In this section, a more technically accurate performance study is conducted, making use of some segmentation quality metrics for quantifying the correctness of the foreground detection provided by algorithm in the traffic.avi standard video in sequence traffic.21, taffic.38, traffic.89 and traffic.98.

In a binary decision problem, the classifier labels sample as either positive or negative. In our context, samples are pixel values, “positive” means foreground object pixel, and “negative” means background pixel.[19] To quantify the classification performance, with respect to some ground-truth classification, the following four basic measures can be used:

True positives (TP): correctly classified foreground pixels;

True negatives (TN): correctly classified background pixels;

False positives (FP): incorrectly classified foreground pixels;

False negatives (FN): incorrectly classified background pixels;

Table 2. 3 Confusion Matrix

	Actual positives	**Actual negatives**
Estimated Positives	TP	FP
Estimated Negatives	FN	TN

The confusion matrix can be used to represent a single point in either receiver operator characteristics (ROC) space or precision recall (PR) space. ROC curves show how the number of correctly classified positive examples varies with the number of incorrectly classified examples, i.e., TPR versus FPR, as given by

$$\text{TP rate} \ : \ \text{TPR} = \frac{\mathbf{TP}}{\mathbf{total\ of\ actual\ positives}} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \qquad (2.\ 16)$$

FP rate: $\text{TPR} = \frac{\text{FP}}{\text{total of actual negatives}} = \frac{\text{FP}}{\text{TN+FP}}$ (2. 17)

Precision: $\text{PR} = \frac{\text{TP}}{\text{total of estimated positives}} = \frac{\text{TP}}{\text{TP+FP}}$

(2. 18)

Recall: RE = TPR. (2. 19)

F-measure: $S_F = 2\left(\frac{\text{PR.RE}}{\text{PR+RE}}\right)$ (2. 20)

Jaccard coefficient: $S_J = \frac{\text{TP}}{\text{TP+FP+FN}}$, $0 \le S_J \le 1$ (2. 21)

Yule coefficient: $S_Y = \frac{\text{TP}}{\text{TP+FP}} + \frac{\text{TN}}{\text{TN+FN}} - 1$, $(-1 \le S_Y \le 1)$ (2. 22)

Finally, *Ilyas et al.*[12] here a weighted Euclidean distance is presented, considering the deviations of FPR and TPR from their respective ideal values 0 and 1. It is defined as follows:

$$E_y = \sqrt{(\gamma\text{FPR})^2 + (1-\gamma)(1-\text{TPR})^2} \quad (2.\ 23)$$

Where γ (0 < γ < 1) is a weighting coefficient, which has to be adjusted according to the desired tradeoff between sensitivity and specificity. In this section, the following index sets have been considered as a valuable quantification of relative performance of each algorithm:

$$S = \{ S_F , S_J , S_Y \} \quad , \quad E = \{ E_{0.25}, E_{0.50}, E_{0.75} \}$$

The first set includes fitness coefficients with an ideal value equal to 1, whereas the second set includes fitness errors with an ideal value equal to 0.

Table 2. 4: pixel based accuracy result for frame difference.

Frame difference (frame no.)	21	38	89	98
PR	0.9552	0.8137	0.8750	0.9003
RE	0.8458	0.9473	0.7459	0.7457
S_F	0.9376	0.8748	0.7723	0.8520
S_J	0.8206	0.7324	0.6064	0.7026
S_Y	0.9369	0.7957	0.9484	0.9755
$E_{0.25}$	0.1201	0.0336	0.2933	0.2069
$E_{0.50}$	0.0797	0.0114	0.2211	0.1506
$E_{0.75}$	0.0271	0.0835	0.1271	0.0772

Table 2. 5 : pixel based accuracy result for Running Average

Running average (Frame no.)	21	38	89	98
PR	0.9199	0.9019	0.7515	0.8736
RE	0.8391	0.9244	0.7329	0.7307
S_F	0.8774	0.9130	0.7420	0.7949
S_J	0.7359	0.7849	0.6728	0.7324
S_Y	0.9038	0.8780	0.7377	0.8508
$E_{0.25}$	0.1260	0.0527	0.2180	0.2189
$E_{0.50}$	0.0846	0.0257	0.1598	0.1612
$E_{0.75}$	0.0307	0.0910	0.0839	0.0849

Table 2. 6: pixel based accuracy result for Simple Statistical Difference (SSD).

SSD (Frame no.)	21	38	89	98
PR	0.9143	0.9626	0.8088	0.9112
RE	0.8731	0.9039	0.7227	0.7099
S_F	0.8932	0.9322	0.7630	0.8307
S_J	0.7572	0.8126	0.6959	0.7756
S_Y	0.9012	0.9331	0.7932	0.9819
$E_{0.25}$	0.0966	0.0701	0.2268	0.2379
$E_{0.50}$	0.0606	0.0393	0.1669	0.1759
$E_{0.75}$	0.0138	0.0995	0.0889	0.0951

Table 2. 7: pixel based accuracy result for Gaussian mixture model (GMM).

GMM (Frame no.)	21	38	89	98
PR	0.8801	0.9280	0.7779	0.9516
RE	0.9972	0.9591	0.9111	0.8332
S_F	0.9346	0.9433	0.8774	0.8880
S_J	0.8171	0.8290	0.7359	0.7501
S_Y	0.8752	0.9090	0.7755	0.9360
$E_{0.25}$	0.0892	0.0226	0.0772	0.1311
$E_{0.50}$	0.0732	0.1010	0.0634	0.0887
$E_{0.75}$	0.0524	0.0750	0.0456	0.0336

Table 2. 8: pixel based accuracy result for Sigma-Delta Background Estimation with Confidence Measurement.

Sigma-delta with con. (Frame no.)	21	38	89	98
PR	0.9912	0.9992	0.8809	0.9697
RE	0.7395	0.7531	0.7337	0.6939
S_F	0.7987	0.8899	0.7996	0.7545
S_J	0.7350	0.7527	0.7380	0.6894
S_Y	0.9438	0.9249	0.8644	0.9198
$E_{0.25}$	0.2988	0.2004	0.2173	0.2383
$E_{0.50}$	0.2256	0.1453	0.1591	0.2578
$E_{0.75}$	0.1302	0.0753	0.0833	0.1530

Table 2. 9: pixel based accuracy result for W4 algorithm

W4 (Frame no.)	21	38	89	98
PR	0.7296	0.8196	0.7127	0.6756
RE	0.9041	0.8175	0.7157	0.7501
S_F	0.8062	0.8655	0.7009	0.7102
S_J	0.7459	0.7201	0.6295	0.6424
S_Y	0.7218	0.7972	0.7734	0.6561
$E_{0.25}$	0.0699	0.0592	0.1329	0.2898
$E_{0.50}$	0.0391	0.0319	0.1719	0.2186
$E_{0.75}$	0.0997	0.0972	0.0926	0.1261

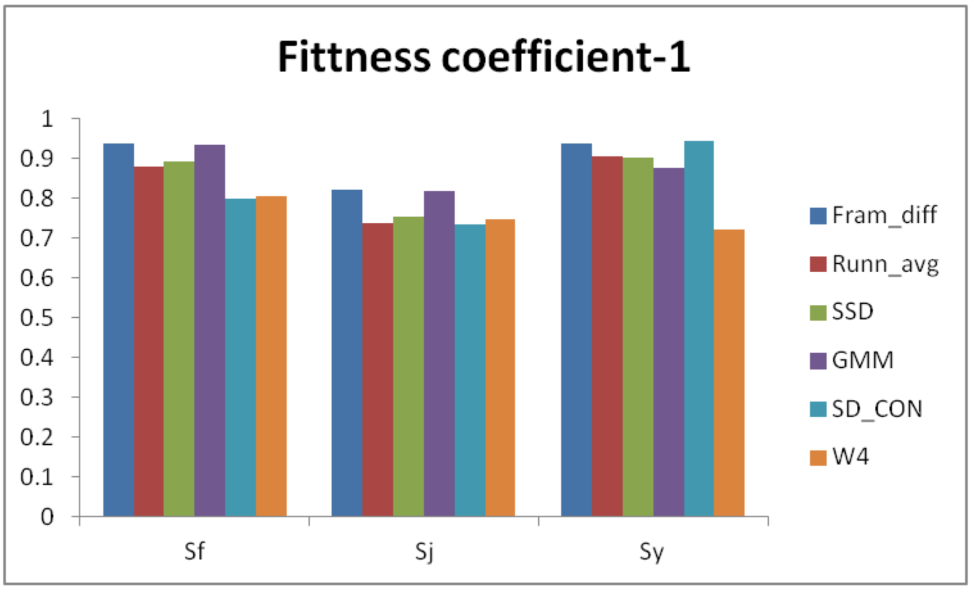


Figure 2. 8: Traffic.21 sequence: Fitness coefficients

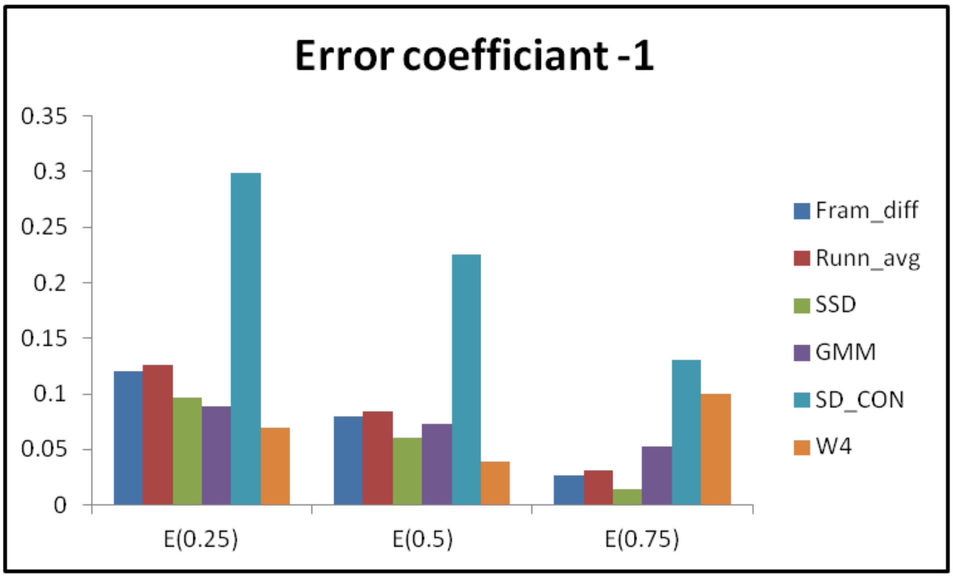


Figure 2. 9: Traffic.21 sequence: Error coefficients

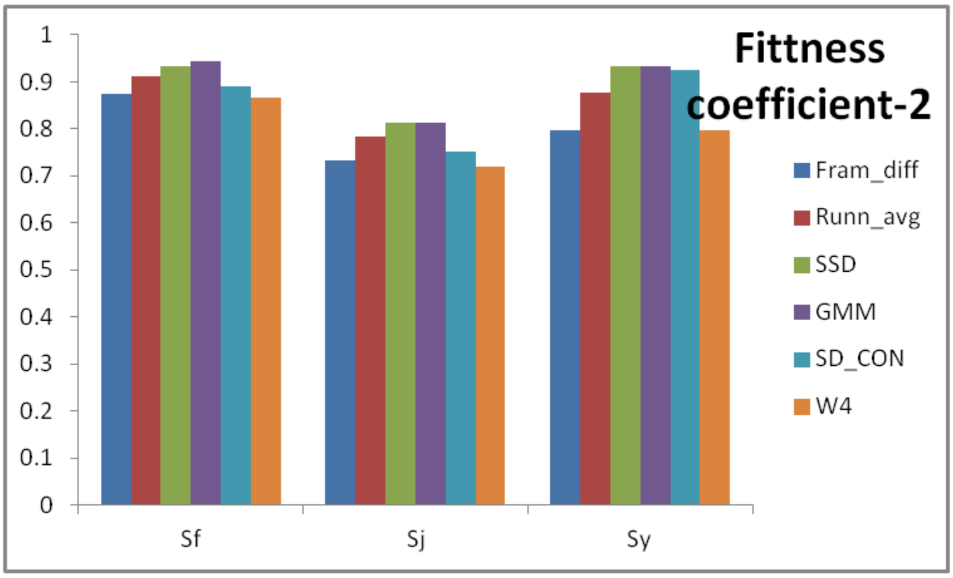


Figure 2. 10: Traffic.38 sequence: Fitness coefficients

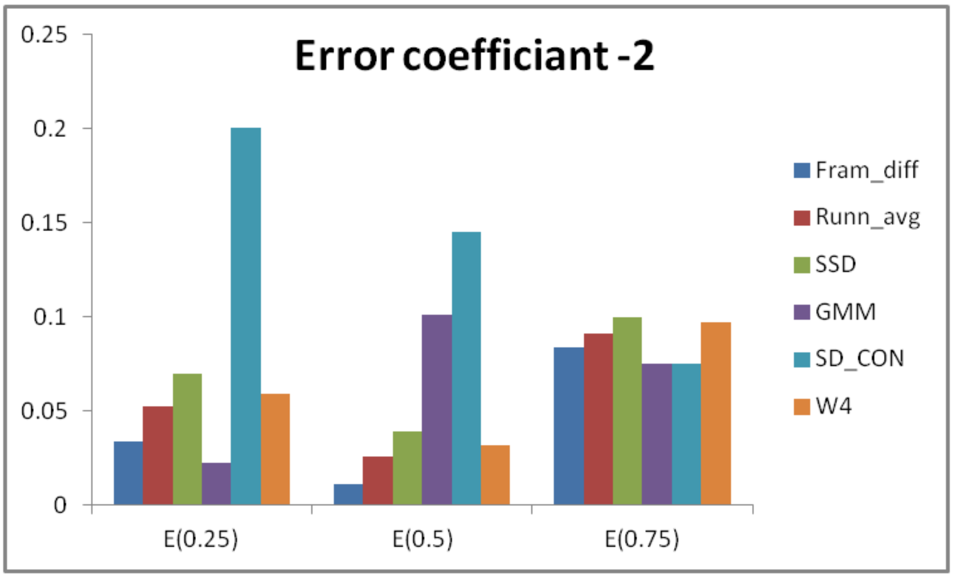


Figure 2. 11: Traffic.38 sequence: Error coefficients

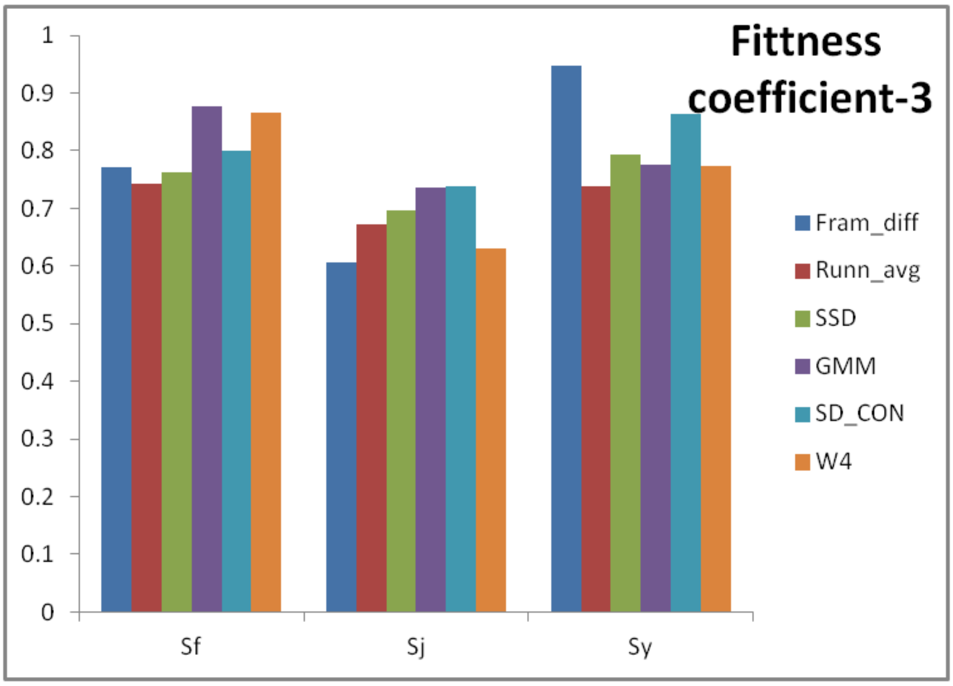


Figure 2.12: Traffic.89 sequence: Fitness coefficients

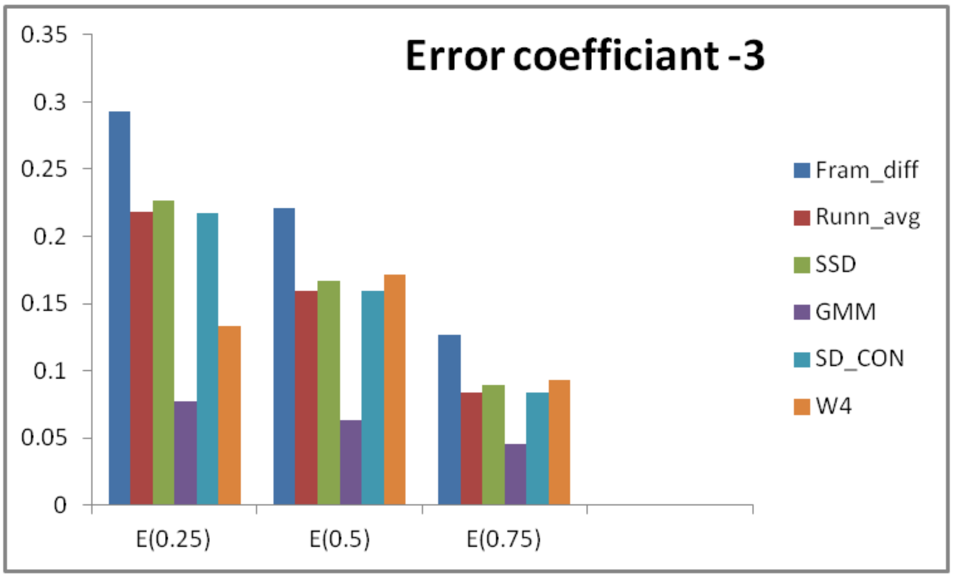


Figure 2. 13: Traffic.89 sequence: Errors coefficients

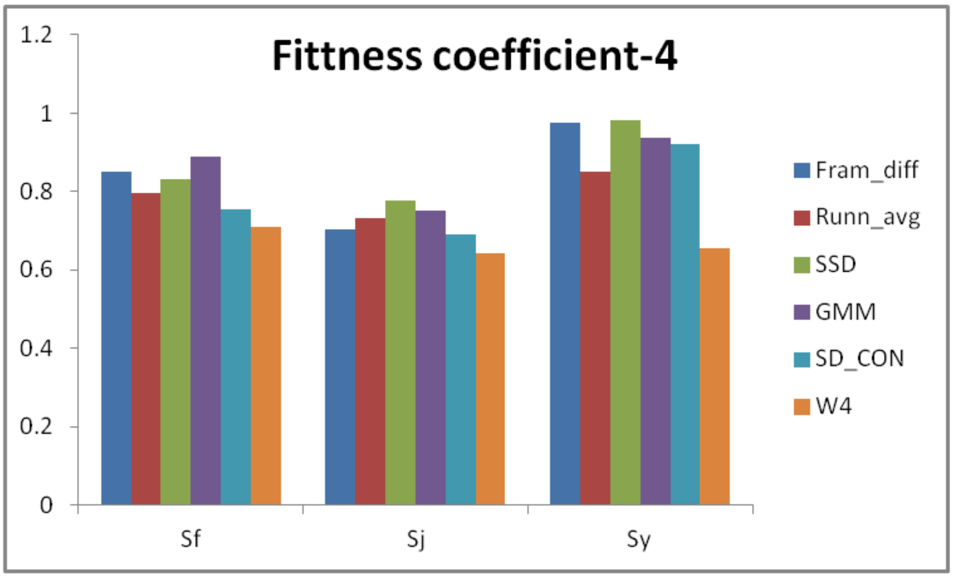


Figure 2. 14 :Traffic.98 sequence: Fitness coefficients

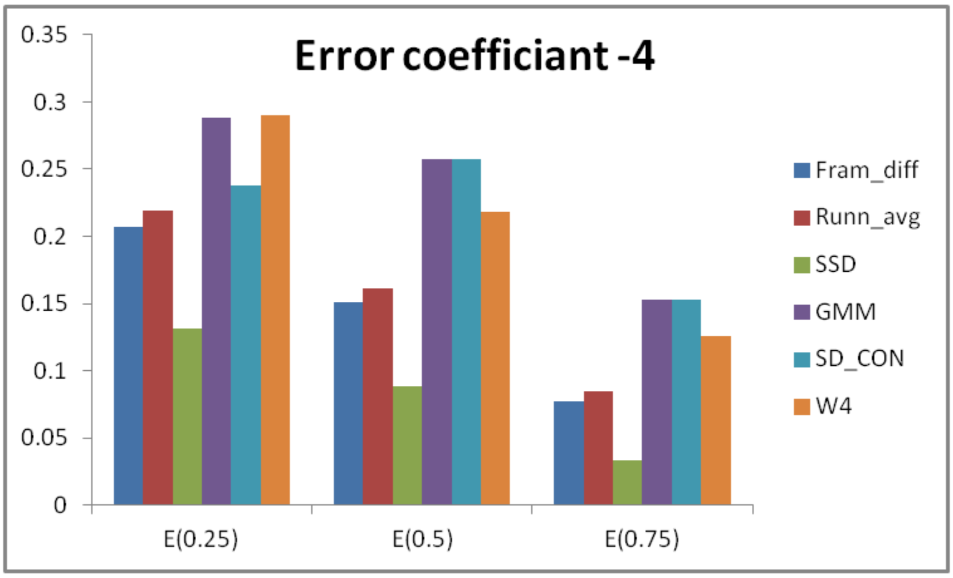


Figure 2. 15: Traffic.98 sequence: Errors coefficients

Results for Campus_raw.avi video has been taken from data base.

Table 2. 10: Results for Campus_raw.avi video

Parameters	Fram_diff	Runn_avg	SSD	GMM	SD_CON	W4
PR	0.7682	0.7132	0.8750	0.8303	0.9432	0.8632
RE	0.6852	0.7394	0.7975	0.8542	0.8321	0.9854
S_F	0.6532	0.6821	0.7372	0.8956	0.9376	0.8524
S_J	0.7542	0.7214	0.7211	0.7324	0.8306	0.9124
S_Y	0.7102	0.7957	0.8612	0.9362	0.9296	0.8594
$E_{0.25}$	0.1286	0.0363	0.1252	0.1305	0.1102	0.1404
$E_{0.50}$	0.2356	0.0141	0.1752	0.1506	0.0977	0.0856
$E_{0.75}$	0.1925	0.0835	0.1986	0.0972	0.0172	0.1117

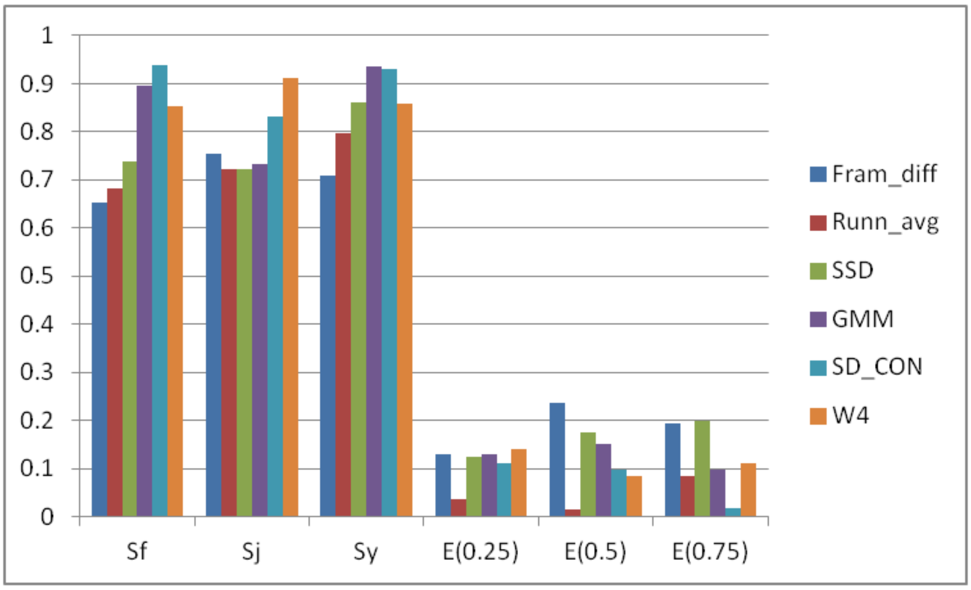


Figure 2. 16 Fitness coefficient and error coefficient for Campus_raw.avi video

Chapter 3

LICENSE PLATE EXTRACTION

License plate recognition (LPR) is one form of ITS (Intelligent Transport System) technology that not only recognizes and counts the number of vehicles but also differentiates them. For some applications, such as electronic toll collection and red-light violation enforcement, LPR records license plates alphanumerically so the vehicle owner can be assessed the appropriate amount of fine. In others cases, like commercial vehicle operations or secure-access control, a vehicle's license plate is compared against a database of acceptable ones to determine whether a truck can bypass a weigh station or a car can enter a gated community or parking lot. [20]

A license plate is the unique identification of a vehicle. The basic issues in real-time license plate recognition are the accuracy and the recognition speed. License Plate Recognition (LPR) has been applied in numerous applications such as automatically identifying vehicles in parking lots, access control in a restricted area and detecting and verifying stolen vehicles. Quality of algorithms used in a license plate detector determines the speed and accuracy of the license plate detection. In the past, a number of techniques have been proposed for locating the plate through visual image processing.

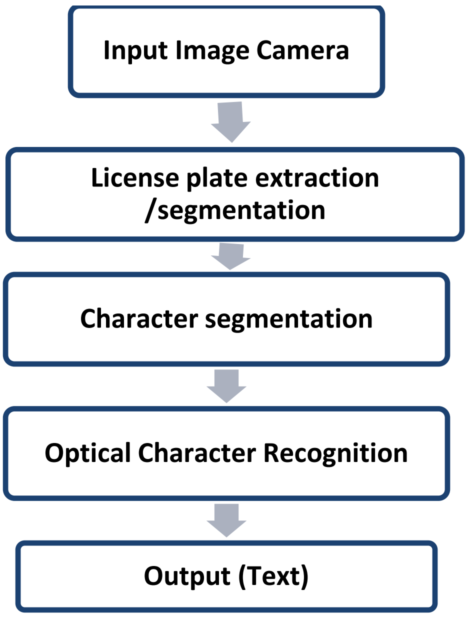


Figure 3. 1Flow chart of license plate recognition

A video is taken from a camera, and then each frame of the video is processed as the image. In this stage the license plate region from the given image is located and isolated. Quality of the image plays an important part hence prior to this stage pre-processing of the image is necessary. So first each frame pre-processed by binarization, noise reduction and edge detection. Then, the license plate is located by different image processing technique.

3.1 Pre Processing

Intensity transformations are applied to enhance the quality of the image for further processing. The following transformations are applied on the colour JPEG image:

- Conversion of RGB image into a Gray-scale Intensity image.
- Conversion of Intensity image into a Binary image.
- Conversion of Gray-scale Image into an edge detected image.

3.1.1 Conversion of RGB image into a Gray-scale Intensity image

MATLAB built-in functions are used for the above transformations.

The image that is acquired from the camera can be an RGB colour image or a Grayscale Intensity image. The algorithm has to check for the RGB image and then has to convert it into a Grayscale image, because all the further processing is done in Grayscale format. Grayscale is chosen because of its simplicity in processing and for its two dimensional matrix nature, and also it contains enough information needed for the actual recognition.

The conversion is performed by using the MATLAB function *rgb2gray*.

3.1.2 Conversion of Intensity image into a Binary image

This transformation, also known as *Image Quantization*, produces a binary image from the intensity image by comparing pixel intensities with a threshold. This stage is very critical for the separation of license plate from the acquired car image. The license plate is assumed to have black characters on a white background and also it is assumed that all the license plates for various cars have approximately the same uniform colour.

Taking the advantage of its brighter background area, the license plate can be separated from the relatively darker car image.

The threshold for binarization can be two types static or dynamic. In static general threshold is taken 150 (in gray scale of 0-255). This value reasonably quantizes those pixels, which represents the license plate and any other portions, which has a pixel-value more than the selected threshold. The remaining portions, which have a pixel-value less than the threshold value, are darkened, as shown in Figure 3.2. In dynamic threshold, the threshold taken will be the average of low and median gray scale values.

The conversion is done by using the MATLAB function *im2bw*.

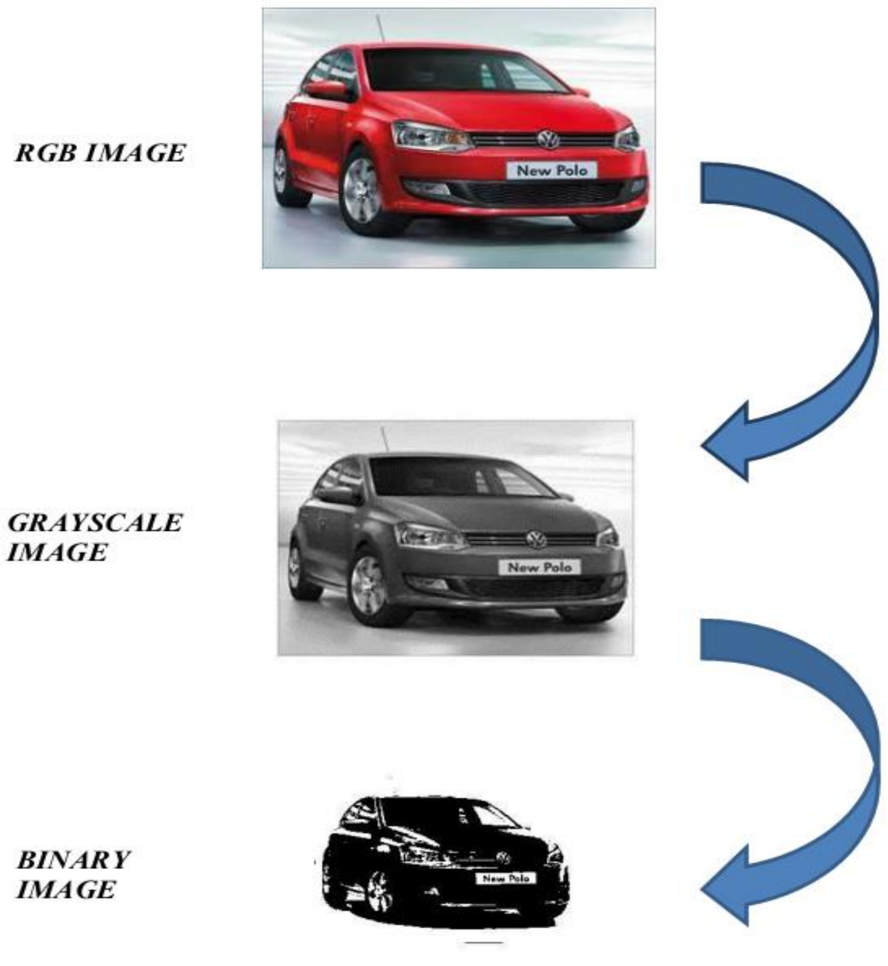


Figure 3. 2: Conversion from RGB Image to GRAY Image, GRAY Image to Binary Image

3.1.3 Conversion of Gray-scale Image into an edge detected image

The Gray scale image is converted into edge detected image. Edge supports six different edge-finding methods as follows

- The Sobel method finds edges of an image using the Sobel approximation to the derivative. It produces edges at those points where the gradient of an image is maximum.
- The Prewitt method finds edges using the Prewitt approximation to the derivative. It gives edges at those points where the gradient of an image is maximum.
- The Roberts method finds edges using the Roberts approximation to the derivative. It produces edges at those points where the gradient of an image is maximum.
- The Laplacian of Gaussian method finds edges by looking for zero crossings after filtering an image with a Laplacian of Gaussian filter.
- The zero-cross method produces edges by looking for zero crossings after filtering an image with a filter you specify.
- The Canny method finds edges by looking for local maxima of the gradient of an image. The gradient is calculated using the derivative of a Gaussian filter. This technique employs two thresholds, to identify strong and weak edges, and comprises the weak edges in the output only if they are connected to strong edges. This method is therefore less prone than others to be affected by noise, and more likely to detect the weak edges.

Out of all these, Sobel edge detection gives the better results. So Sobel edge detection is taken.The conversion is performed by using the MATLAB function *edge.*

3.2 License Plate Extraction

Once the Pre-processing is done there are many ways to extract the License plate. They are

- Hough Transform
- Template matching

- Region growing
- Histogram Approach

3.2.1 Hough transform

In Hough transform approach, the first step is to threshold the Gray scale source image. Then the resulting image is passed through two parallel processes for the extraction of horizontal and vertical line segments respectively.

The first step in both of these sequences is to extract edges. The result is a binary image with edges highlighted. This image is then used as input to the Hough transform, which produces a list of lines in the form of accumulator cells. These cells are then analysed and line segments are computed.

Finally the list of horizontal and vertical line segments is combined and any rectangular regions matching the dimensions of a license plate are kept as candidate regions. This also gives the output of the algorithm.

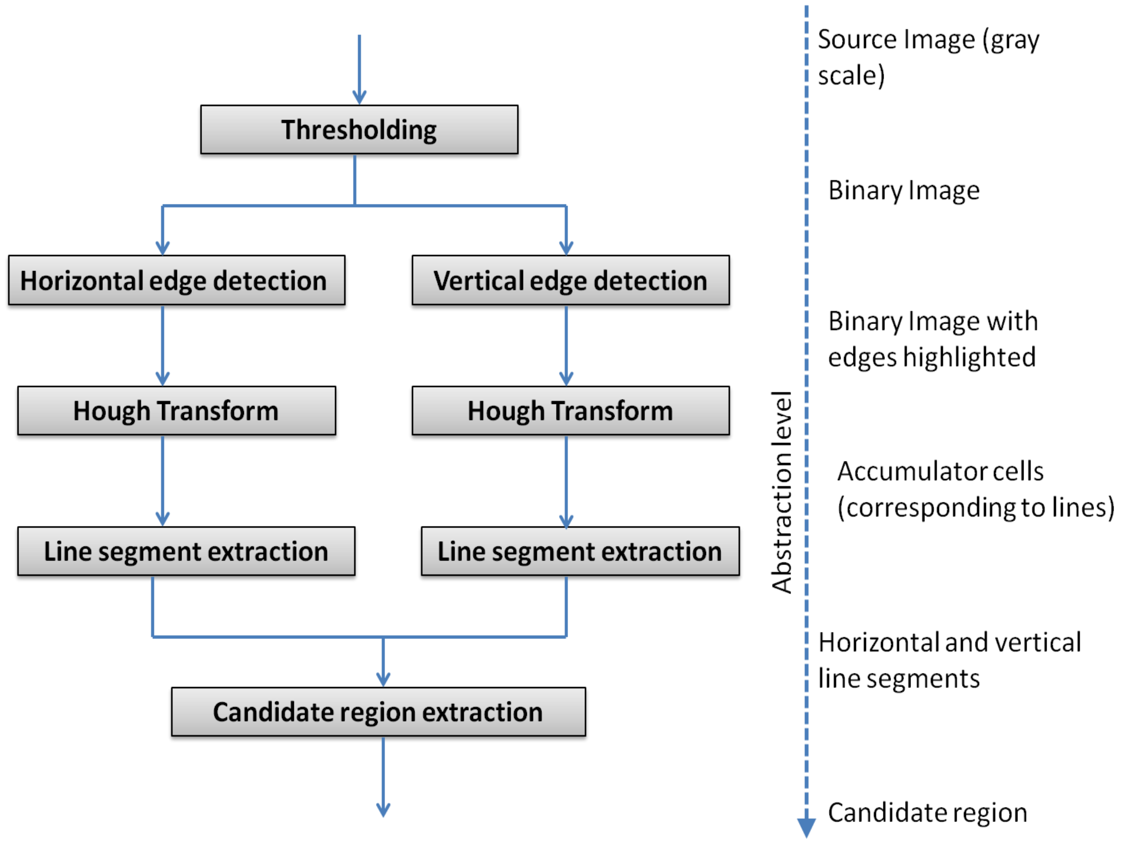


Figure 3. 3: Hough Transform Approach

Advantages	Explanation
Scaling invariant	Since the algorithm does not look for regions of particular size, it is invariant to scaling of the license plate
Relatively independent of license plate colour	As long as the license plate is brighter than the surroundings, the plate is usually correctly extracted.

Limitations	Explanation
Trouble detecting vertical lines	Vertical lines in the license plate are typically more than a factor four shorter than the horizontal lines and thus more susceptible to noise.
Finds more than just the license plate	All rectangular regions with dimensions equal to that of a license plate are identified, which is sometimes many. This makes it difficult to choose the correct candidate later.

3.2.2 Template Matching

The main concept behind extraction through template matching is, that by comparing each portion of the investigated image to a template license plate, the actual license plate in the image is found as the region bearing the most resemblance to the template.

The way the template is constructed plays a significant role in the success of template matching. There are no rules for template construction but the template must be constructed in such a way, that it has all the characteristics of a license plate as it is presented in the source image.[25]

When examining the source images in the training set, it is quite obvious that there are two major differences in the plates. The plates vary in size because of the fact that the cars are at different distances and the license plates vary in light intensity due to different lighting conditions when the images are taken. Both are issues that cannot be handled by the template construction. The size issue cannot be helped at all. This observation might very well turn out to be critical for this approach, but for now that fact will be ignored and the attention turned to the lighting issue. This can be helped by proper image pre-processing.

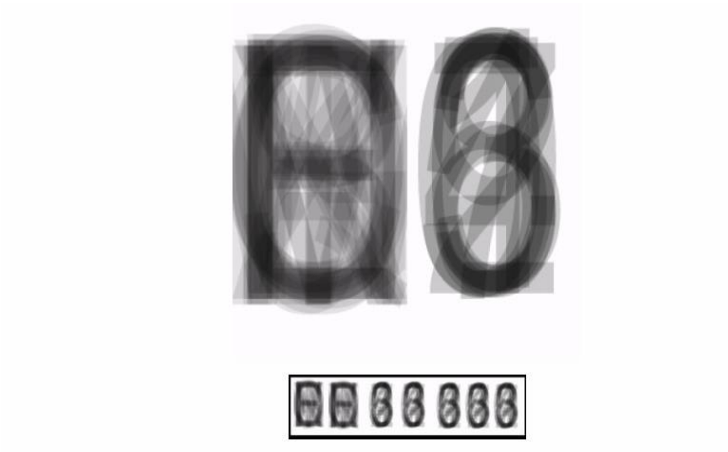


Figure 3. 4: Template created through opacity-merging of the probabilities

The left hand side shows an alphabet template and the right hand side a digit template.

Advantages	Explanation
Single and simple similarity measure	Template matching is a strong approach or finding a single similarity measure between two images. Through a simple off-the-page algorithm this measure can be easily calculated with good results without an investigation of the specifics of the region sought after. Often, using a sample region will be sufficient to identify similar regions

Limitations	Explanation
Slow algorithm	A large input image and a smaller template will make performing the simple

	calculations in the many nested summations a demanding task.
Not invariant to rotation and perspective distortion	If the region sought after is rotated or distorted in the input image, the region may very well bear little or no resemblance to the template on a pixel by pixel basis. This means the similarity measurement will fail.
Not invariant to scaling	Scaling of input images proves to be an unsurpassable problem. It is an impossible task to examine the input image using all possible sizes for the template, and even the smallest variation in size will often lead to a wrong result.
Static threshold	The images vary a great deal in overall brightness, depending on the surroundings.

3.2.3 Region Growing

The basic idea behind region growing is to identify one or more criteria that are characteristic for the desired region. Once the criteria have been established, the image is searched for any pixels that fulfil the needs. Whenever such a pixel is encountered, its neighbours are checked, and if any of the neighbours also match the criteria, both of the pixels are considered as belonging to the same region. Figure 3.6 visualizes these steps. The criteria can be static pixel values, or depend on the region that is being expanded.

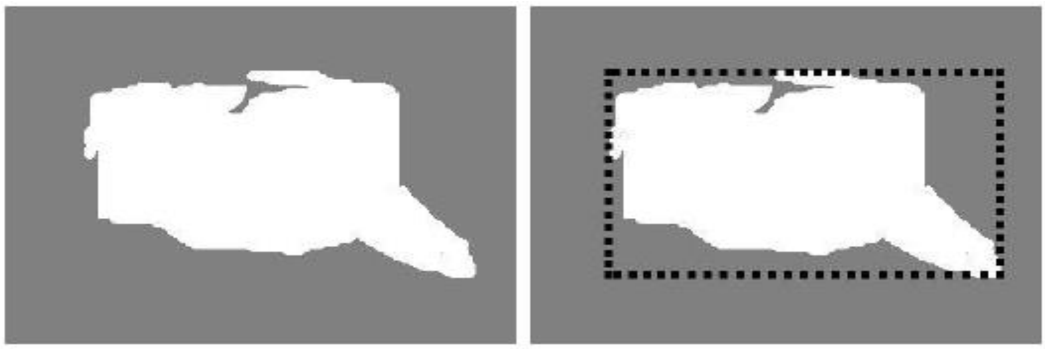


Figure 3. 5: Largest contained Rectangle

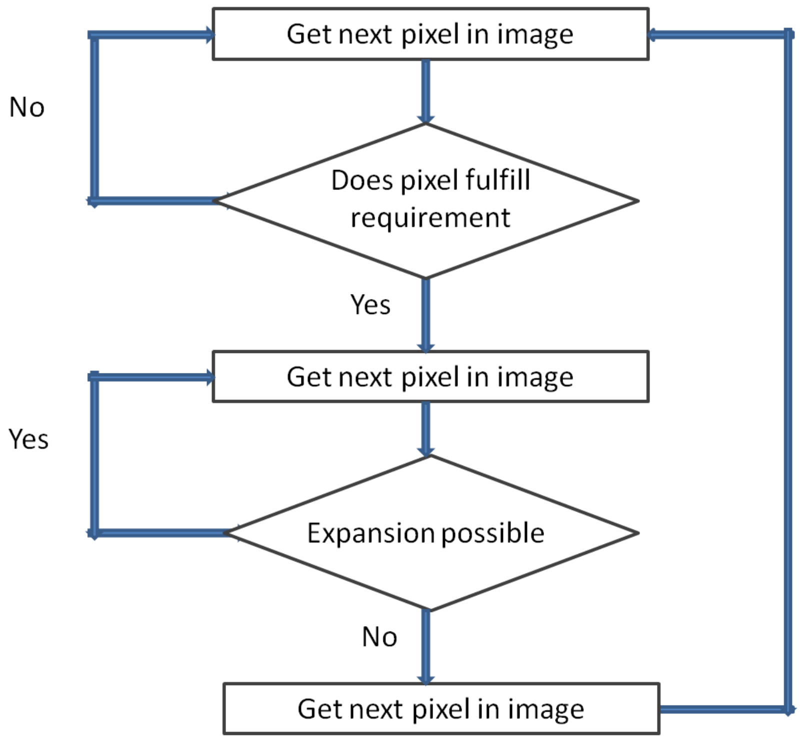


Figure 3. 6: Steps for Region Growing

Advantages	Explanation
Fast algorithm.	Each pixel is examined no more than once for each neighbour.
Invariant to distance between camera and Vehicle	Candidates with the correct shape are extracted by this method and it does not depend on size of regions.

Limitations	Explanation
High demands for memory	The recursive nature of the algorithm stores temporary results for each call to the recursive function.
Static threshold	The images vary excessively in overall brightness, depending on the surroundings.

3.2.4 Histogram approach

For extracting the license plate, one approach is to take row wise histogram and using threshold we can detect the boundaries. And out of the extracted regions which will satisfy the dimensions of license plate. This approach is shown in Figure 3.7.[25]

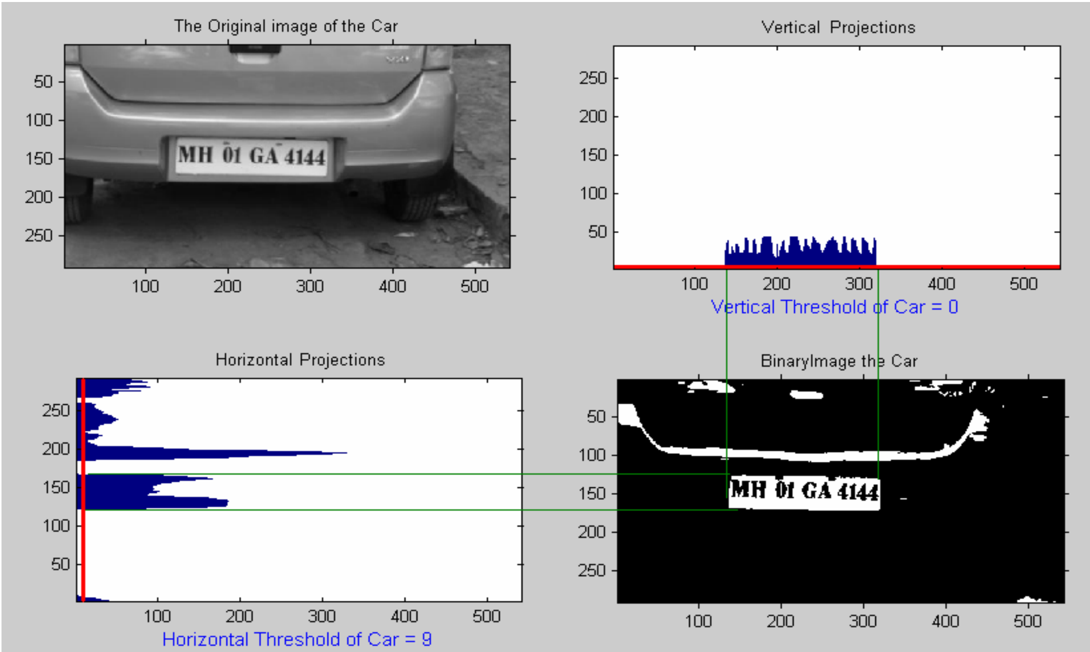


Figure 3. 7: Histogram Approach

3.3 Proposed Method I for license plate extraction

This Technique is based on observation that “The License plate is the noisiest part of the Car Image”. It means that if the edges of the image are taken, then we get the most edges in the License Plate Area. Here we used Sobel edge detection Technique to detect the edges of the image. After detecting the edges we get a binary image. Now a window is run over the image and no of white pixels are counted. And when white pixels are 10% – 30% of the total pixels, we take that region may contain a License plate. From the available plates we check for the number of connected objects in that region. If this count is in between 8-15, we consider that as a license plate.

The entire approach is shown in Figure 3.8 flow chart view.

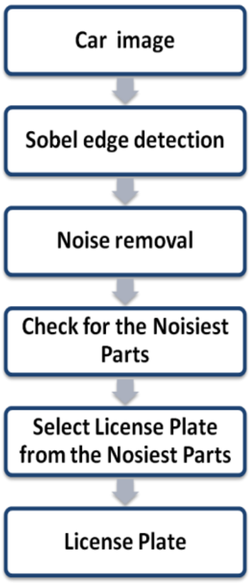


Figure 3. 8: Flow Chart of the Method

3.3.1. Sobel edge detection

Given the binarized image we perform vertical edge detection with *mask A* and horizontal edge detection with *mask B*. Since we are not interested in the direction of the edge, we take the absolute value of the output of the mask to obtain edges present in all four directions. Wherever an edge is present we mark the pixel as a 1, otherwise, we mark it 0.

If the binarization threshold is appropriate, performing edge detection on the black and white image should result in a great deal of edges in the area of the license plate due to the characters. This is a property we can use to choose possible license plates. The 3x3 mask used to detect edges that are relatively thin comparing to the lines of the characters on a license plate. Therefore, the edge detector will detect much thinner edges in addition to the edges of the characters of the license plate. Construct masks that specifically detect lines that are roughly the thickness of the characters on a license plate to minimize false alarms.

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \qquad\qquad \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

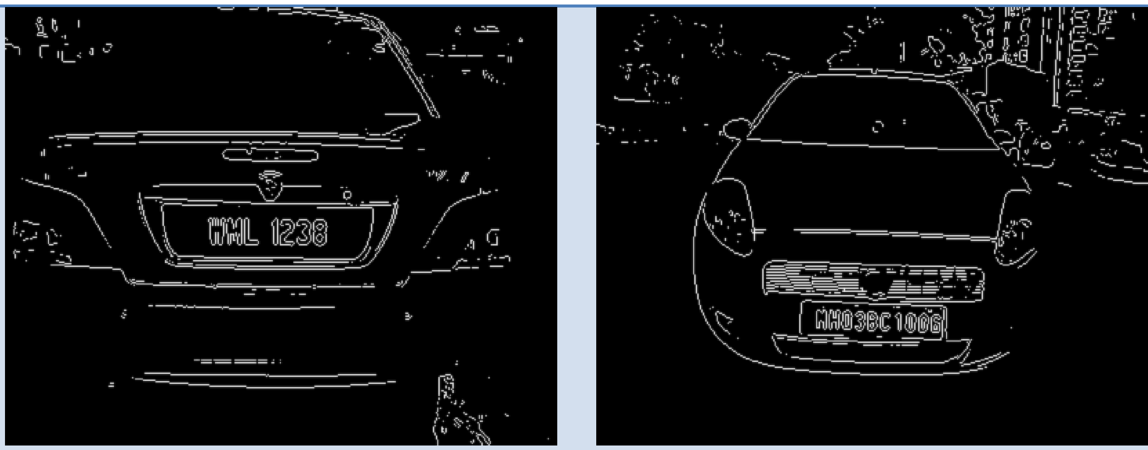
Mask A Mask B

CAR Image-1	CAR Image-2

(a)	(b)

Figure 3. 9: Car Image (a) CAR-1 (b) CAR -2

CAR Image-1	CAR Image-2



(a)	(b)

Figure 3. 10: Sobel edge detected Images (a) CAR-1 (b) CAR-2

3.3.2. Removal of unwanted small connected component

As it can be observed in Figure 3.10, there will be many unwanted small connected component or we can say noise parts in the image such as screws and nuts and small dust particles, which may give the false alarm. Sometimes shadows of the leaves also create the wrong results. To overcome this problem, noise is removed after the edge detection. To achieve this connected pixels are counted and if the number of connected pixels are less than the specified value, then that is taken as the noise. This step drastically increased the accuracy of the system. If the Threshold is selected high, and then the characters may be eliminated. If the Threshold is taken less, then the noise may not be removed perfectly, so optimum threshold must be selected. Noise removed image is shown in Figure 3.11.

3.3.3. Checking for the Noisiest Parts

After removing the noisy parts, a window is run over the image so as to find the license plate area. License plate area will contains 10 – 30% of white pixels in the binary image.

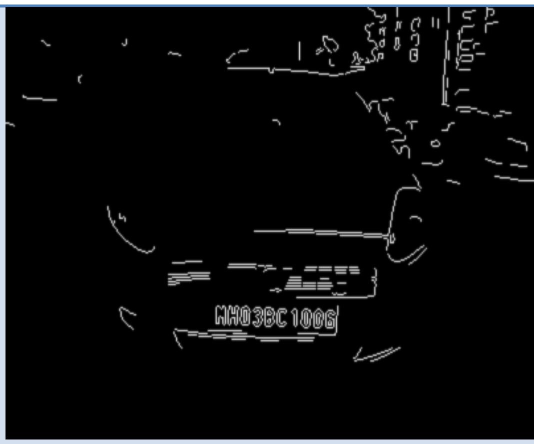
CAR Image-1	CAR Image-2
	
(a)	(b)

Figure 3. 11: Noise removed images (a) CAR-1 (b) CAR-2

CAR Image-1	CAR Image-2
	
(a)	(b)

Figure 3. 12: License Plate Area

3.3.4. Select License Plate from the Nosiest Parts

After selecting the possible license plates, actual license plate can be found by region growing technique. In our case license plate is taken as the window, which contains maximum number of the white pixels in it. The selected License plate area is shown in Figure 3.12.

Advantages	Explanation
Invariant to distance	As this algorithm checks for noisy parts, it is invariant to size; nevertheless the license plate should not exceed the window size.
Noise tolerance	As we are calculating vertical edges there may be some small lines present in the edge detected image. But as we remove small connected parts like screws, noise will not affect the system much.
Rotation invariant	As we are finding the area which contains license plate, rotation of the plate will not affect the algorithm. But to make the license plate straight, we had to use Hough transform at later stages.
Colour Invariant	This method is invariant to foreground and background colours of License plate.

Limitations	Explanation
Slow	Large input image and less window size slows down the algorithm even then the time complexity of the system is O (n).
Find more than just the license plate	As it gives the optimum window there will be some other components present, which have to be filtered in later stages.

3.4 Proposed Method- II for license plate extraction

Here we are proposing Block Variance Technique for license plate extraction. The license plate having alphanumeric characters with high variance as compared to rest part of the image, so we are using this feature of license plate for extracting it from the image.

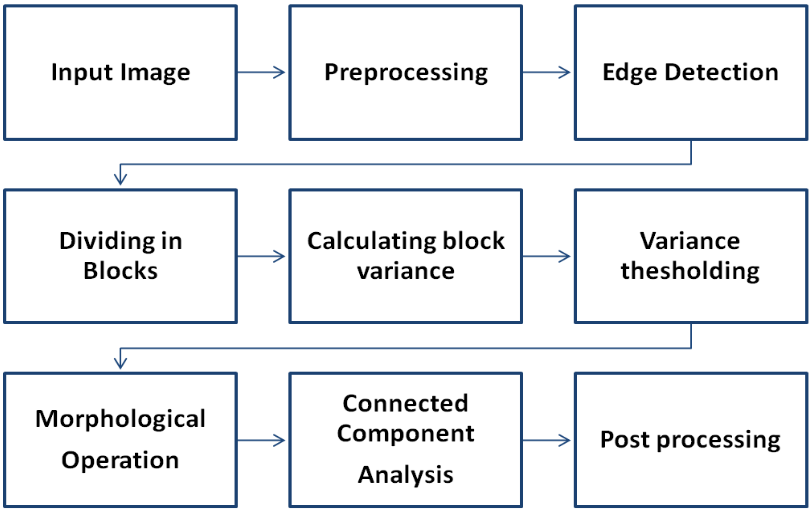


Figure 3. 13: Flow chart of Block Variance Technique

3.4.1 Pre-processing

In pre-processing stage some transformations are done. They are

- Conversion of RGB Image into a Gray-scale image.
- Resizing all input images in a pre-defined size.
- Converting 8 bit image to 4 bit image

First we have to convert RGB image to Gray scale image, because we are interested only in intensities value of the images. MATLAB inbuilt function *rgb2gray* is used for it.

In block variance technique we are dividing image into fixed size blocks so for this all the input images should be of same size. MATLAB inbuilt function *imresize* is used to resize the input image into a fixed resolution.

3.4.2 Morphological Operation

Morphological operations are a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, and are especially suited to the processing of binary images. Morphological techniques analyse an image with a small shape or template called a structuring element. The structuring element is placed at all possible positions in the image and it is compared with the neighbourhood pixels. Some operations test whether the element fits within the neighbourhood, while others test whether it intersects (hits) the neighbourhood: [23]

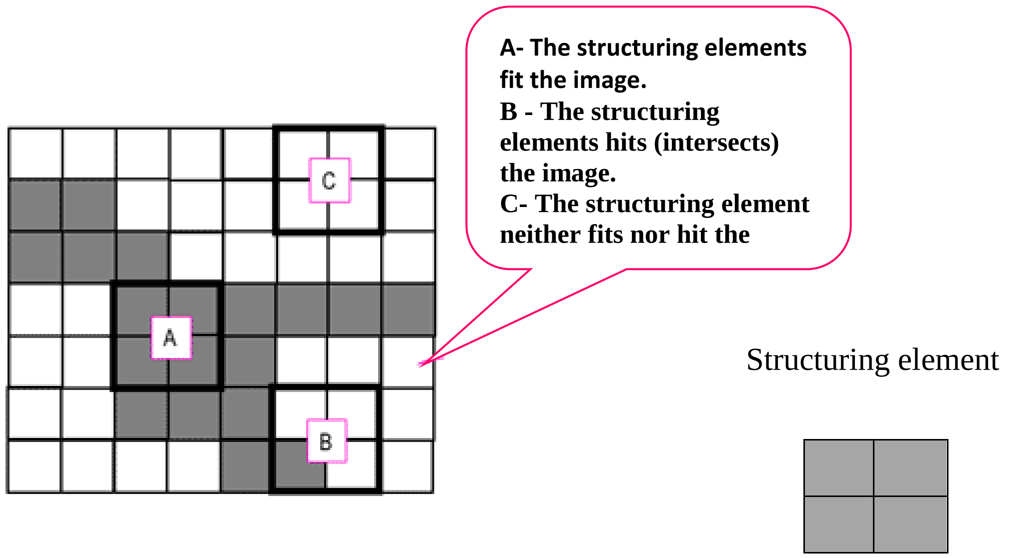


Figure 3. 14: Probing of an image with a structuring element
(White and Gray pixels have zero and non-zero values, respectively)

The structuring element is a small binary image, i.e. a small matrix of pixels, having values zero and one.

Fundamental Morphological operations

Erosion and dilation

The erosion of a binary image *f* by a structuring element *s* (denoted $f \ominus s$) creates a new binary image $g = f \ominus s$ with ones in all locations (x, y) of a structuring element's origin at which that structuring element *s* fits the input image *f*, i.e. $g(x, y) = 1$ is *s* fits *f* and 0 otherwise, repeating for all pixel coordinates (x, y).

The dilation of an image f by a structuring element s (denoted $f \oplus s$) produces a new binary image $g = f \oplus s$ with ones in all locations (x, y) of a structuring element's origin at which that structuring element s hits the input image f, i.e. $g(x, y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x, y). Dilation has the opposite effect to erosion. It adds a layer of pixels to both the inner and outer boundaries of regions. [22]

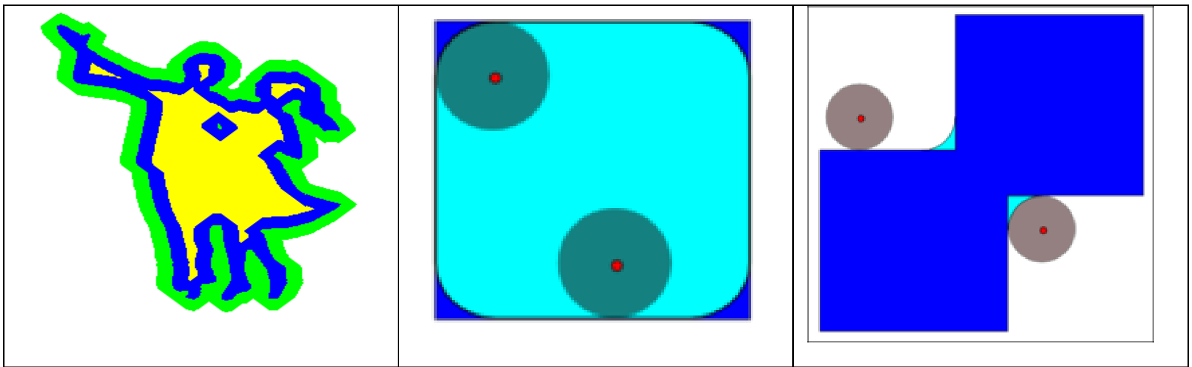
Opening and Closing

The opening of an image f by a structuring element s (denoted by $f \circ s$) is erosion followed by dilation. Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels.

$$\text{Opening:} \qquad f \circ s = (f \ominus s) \oplus s \qquad (3.1)$$

The closing morphological operation performed on an image f by a structuring element s (denoted by f • s) is a dilation followed by erosion: Closing is so called because it can fill holes in the regions while keeping the initial region sizes.

$$\text{Closing:} \qquad f \bullet s = (f \oplus s) \ominus s \qquad (3.2)$$



(a) (b) (c)

Figure 3. 15 :(a) Erosion and Dilation (b) Opening (c) Closing.

In Figure 3.17 (a) A shape (in blue) and its morphological dilation (in green) and erosion (in yellow) by a diamond-shape structuring element. (b) The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners. (c) The closing of the dark-blue shape (union of two squares) by a disk, resulting in the union of the dark-blue shape and the light-blue areas.

3.4.4 Block Variance Algorithm

step 1: First an image is taken, then pre-processing is done to remove unwanted noise and to increase the image contrast.

step 2: Resizing of all input image in 300 x 400 pixels because we have to divide the whole image into the blocks so all the images should be of same pixel resolution.

step 3: Converting RGB image into Gray scale image because we only interested in intensity values.

step 4: The license plate of the car consists of several characters, so the plate area contains rich edge information. We used Sobel edge detection Technique to detect the edges of the image.

step 5: Dividing image in (5 x 5) blocks i.e. total 25 blocks of 60 x 80 pixels then finding variance of each block.

step 6: Calculating threshold variance using (3.3).

$$V_{th} = \frac{V_{max}+V_{min}}{2} \qquad \text{..............} \; (3.3)$$

Where V_{max} and V_{min} are the maximum variance and minimum variance respectively.

step 7: Segmenting all the blocks by V_{th} , blocks having low variance as compare to threshold variance are removed.

step 8: After that we are removing the long lines.

step 9: Performing morphological closing operation with rectangular structuring element of size [2, 20].

step 10: Then using connected component analysis for labelling the each component and extracting the license plate using region properties.

step 11: After extracting the number it will converted to binary image, enhancing image resolution and then sending this image to character segmentation module for further process. Results are shown in figure 3.16

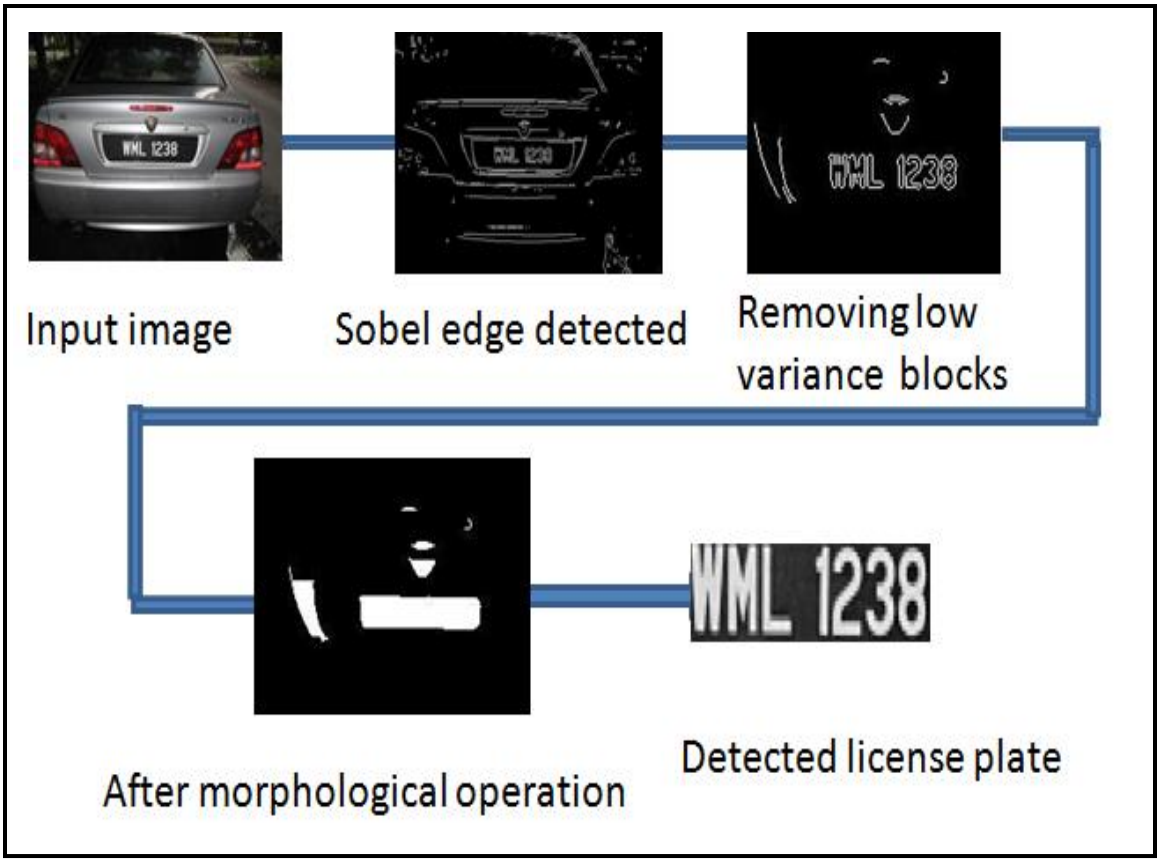


Figure 3. 16: Results of Proposed Block Variance Algorithm

Here (a) shows the input image CAR-1 and CAR-2 (b) image after applying sobel operator (edge detected image) (c) removing low variance blocks (d) after morphological operation (e) showing connected component (f) detected license plate

3.5 Results and Discussion

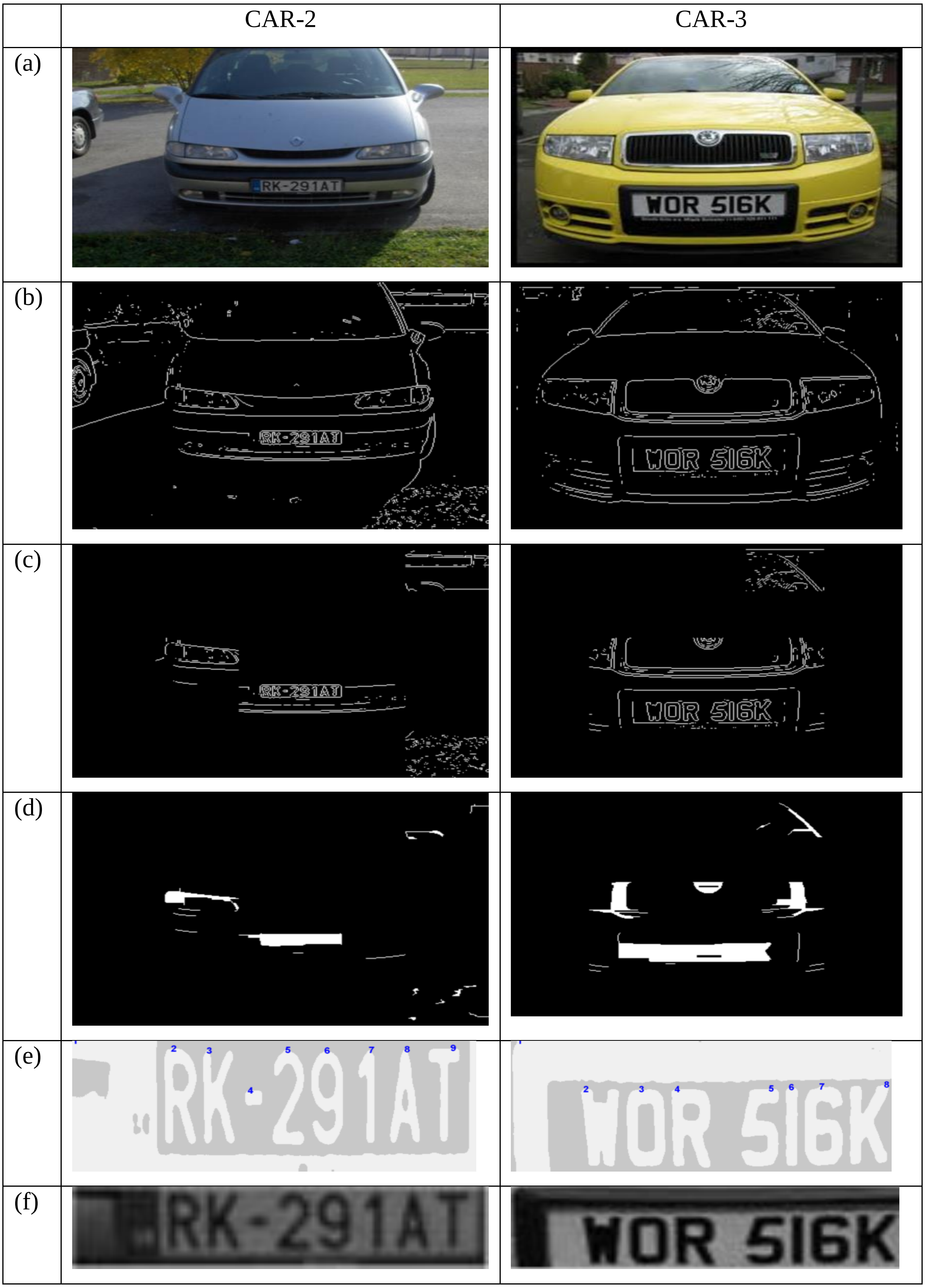


Figure 3. 17: Result of block variance technique for CAR-2 and CAR-3

	Car image	Detected number plate(gray image)
1.		
2.		
3.		
4.		

Figure 3. 18: Some more results for block variance technique.

We have tested the algorithm for data base that has been taken from Video and Image Processing laboratory data set 128 x128 ppm car images of different countries. Total 90 images the algorithm has been tested including all types of license plates like plates having different background colour, different size, different lighting condition, standardized license plate, license plate with good contrast, low resolution license plate and skewed license plate.[21]

3.5.1 Measure of Effectiveness

Capture rate represents the ability to locate and identify a license plate for block variance technique.

$$capture\ rate = \frac{number\ of\ license\ plate\ correctly\ recognized}{total\ number\ of\ test\ images\ taken} \tag{3.3}$$

Table3. 1Capture Rate for license plate detection

Type of license plates	Number of images	Correctly recognized license plates	Capture rate in %
Standard license plates	20	19	95
License plates with good contrast	15	15	100
License plates with Proper lighting condition	15	14	93.33
License plates with Low resolution	15	12	80
Skewed License plates	10	7	70
Non-standard size of license plates	15	13	86.66

Overall capture rate for the algorithm

Capture Rate = 87.49%

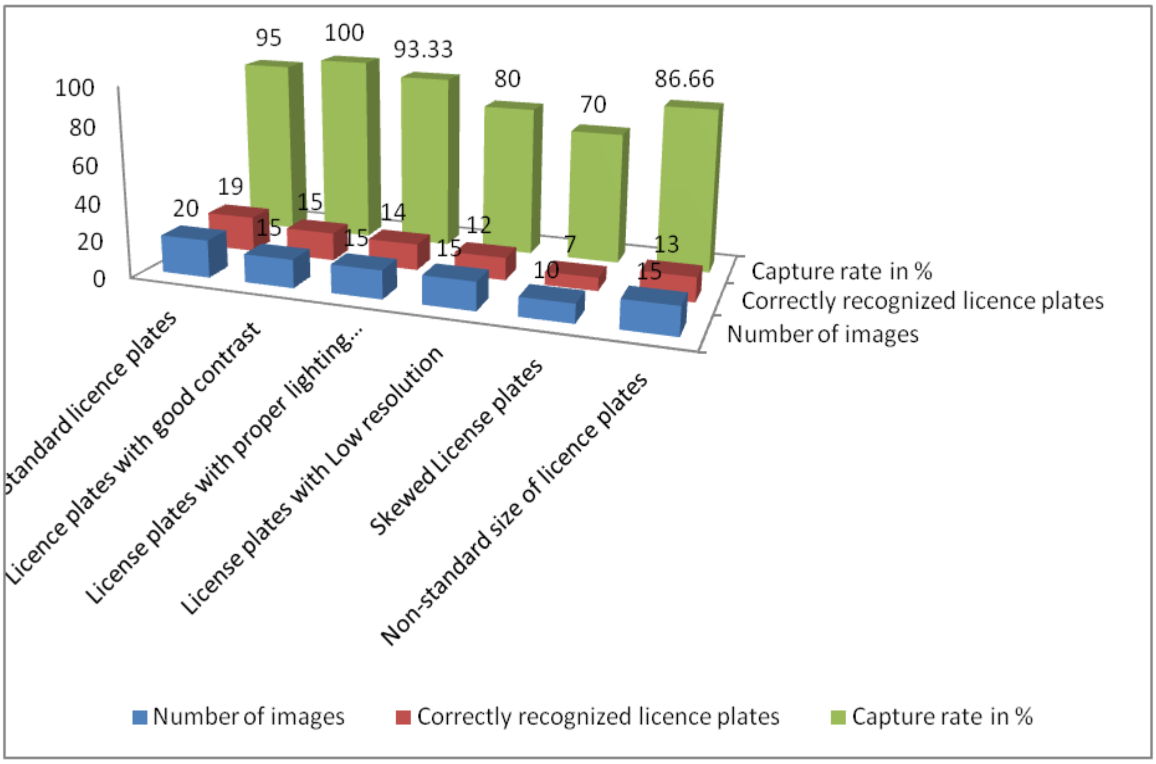


Figure 3. 19 : Measure of effectiveness

3.6 Comparison

Comparison of the above methods is shown in the following table.

Table3. 2Comparison of different license plate extraction method

Area	Hough Transform	Template Matching	Region Growing	Histogram Approach	Edge Detection	Block variance Technique
Scale Invariance	YES	NO	YES	Within Limits	Within Limits	**YES**
Lightening invariance	YES	NO	NO	YES	YES	**YES**
Rotation Invariance	NO	NO	YES	NO	YES	**YES**
Colour Invariance	YES	NO	NO	NO	YES	**YES**

Chapter 4

Character Extraction

Character Extraction or character segmentation is the important component of our recognition system. It takes a properly segmented license plate as an input. Some pre-processing (Morphological operators) is done on the license plate image for the removal of noise and the noise free output image is sent for character segmentation. Image binarization and image projections are used for character extraction. Character segmentation is an important step in License Plate Recognition system. There are many difficulties in this step, such as the effect of image noise, space mark, plate frame, rivet and so on.

The spots remaining after the previous stage are arranged in the form of a string and are treated as possible license plate characters. Each of these candidate characters is size-normalized to a reference size, before template matching against a set of stored templates is performed.

4.1. Character Extraction

For isolating Characters we assume that the license plate is horizontal and we proceed.

There are two ways to extract the characters of License Plate.

- Histogram approach
- Connected pixels method

4.1.1 Histogram approach

Two histograms are computed for the segmented license plate image - the first one along the X-axis, and the second one - along the Y-axis, as shown in the Figure 4.1. These histograms are obtained by using the same techniques that are used for plate localization. In Y-histogram a uniformly distributed pattern of low-valued breaks between characters is detected, whereas in X-histogram the top and bottom borders of the character set are identified. [17]

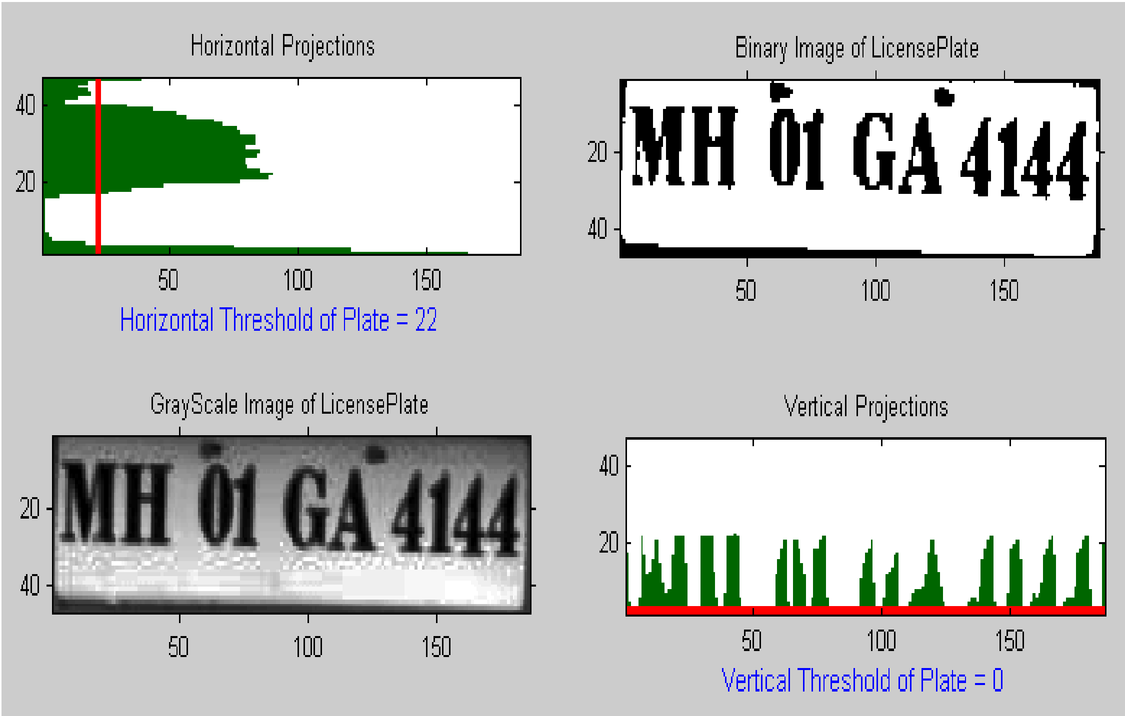


Figure 4. 1: Histogram approach for character Recognition

4.1.2 Connected pixels approach

In this method we first take the negative of the image and then find out the connected objects in that. And there are many inbuilt functions for finding connected pixels in MATLAB. As shown in Figure 4.2

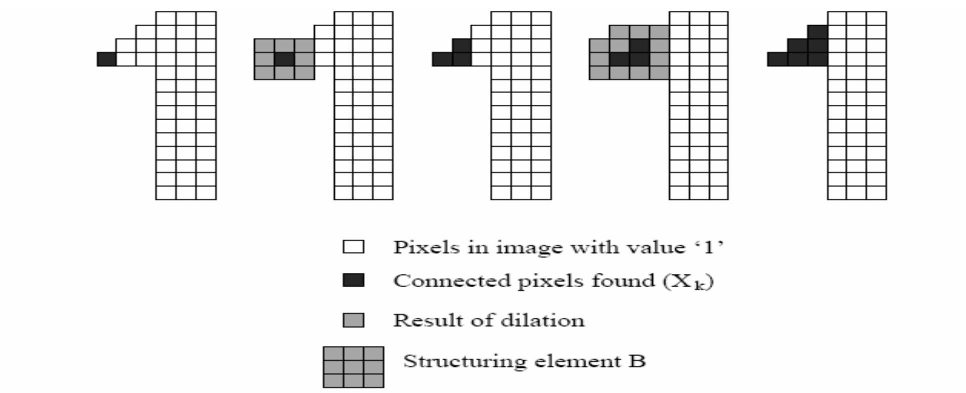


Figure 4. 2: Connected pixels approach for character extraction

4.2 Character Recognition Using Template matching based OCR

The OCR technique is used in order to recognize different digits. This approach is based on pattern recognition principles. The system of OCR engine is based on a template-matching algorithm.

This method basically matches the linear relationship between the detected character image and the standard database images of characters. The concept of correlation coefficient is used for template matching. The correlation coefficient is a measure of the strength of the straight-line or linear relationship between two variables and it takes values ranging in between 0 and 1. Formula for the computation of correlation co-efficient is

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \tag{4.1}$$

Here we are getting extracted license plate from the previous section of the system so first pre-processing is performed.

4.2.1. Pre-processing

Pre-processing is one of the preliminary steps in character recognition. Before the raw data is used for feature extraction it has to undergo certain preliminary processes so that we get accurate results. This step helps in correcting the deficiencies in the data which may have occurred due to the limitations of the camera sensor. The input for a system may be taken in different environmental conditions. Same object may give different images when taken in different time and conditions. Hence by doing pre-processing we get data that will be easy for the system to operate on, there by producing accurate result. Preprocessing evolves image sharpening, enhancing image contrast, increasing the resolution of the image and converting grayscale license plate to binary images.

4.2.2. Flow Chart of template matching algorithm

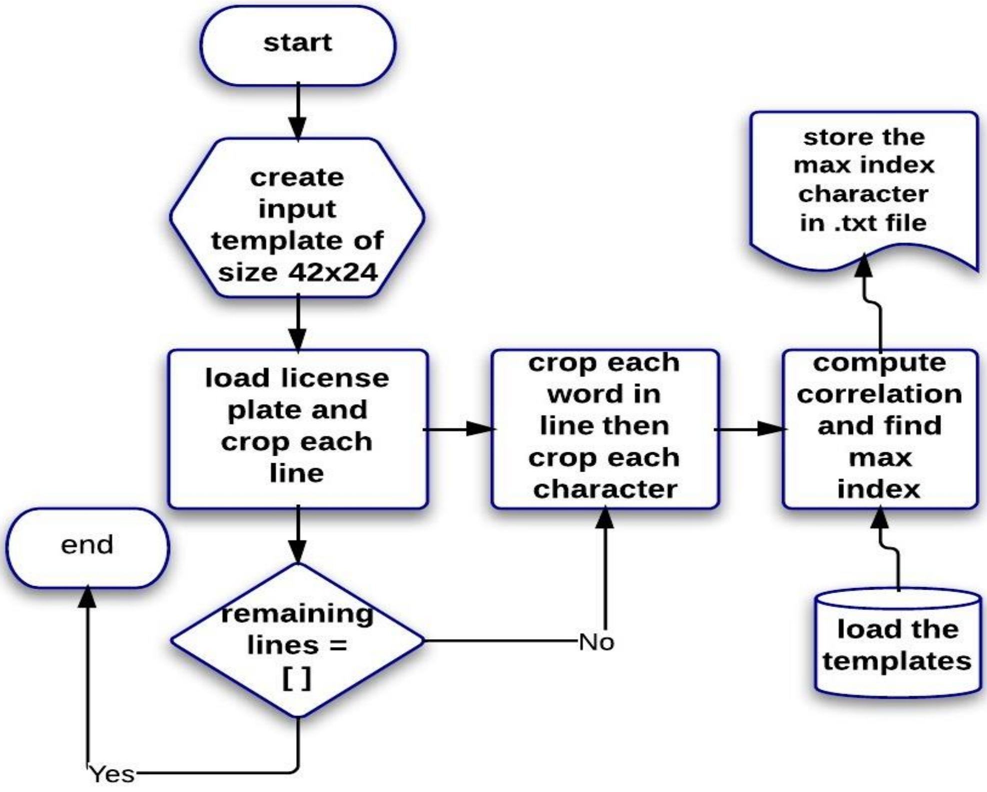


Figure 4. 3: Flow Chart of Template Matching

The following steps have been used to recognize the character from the extracted license plate using template matching algorithm.

step 1: First step is to create templates (A-Z), (0-9) of size 42 x 24 (binary image). It is important that all the template characters created should be of same window size.

step 2: The extracted license plate image is first loaded and pre-processing is performed.

step 3: After converting the input image to binary, each character from the image is detected by the technique of segmentation. First, the image is cropped to fit the text. After that, line by line the image is cropped.

step 4: Thereafter in each line word by word the image is cropped, followed by cropping of each character in a word to fit the text.

step 5: Each character which is detected is resized to the size of template window (42x24).

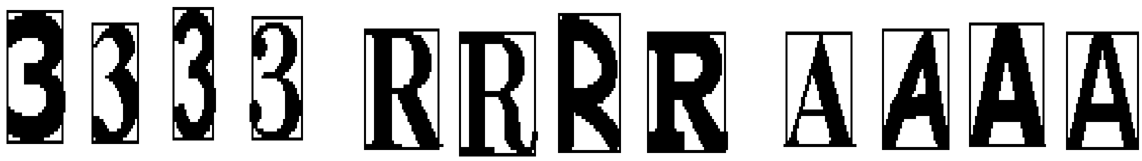
step 6: After resizing the character, correlation coefficient for each template with the character is found and the correlation coefficient values are stored in a matrix.

step 7: The main operation used for the classification was the two-dimensional correlation. This operation gives a value of the similarity between two matrices (images).

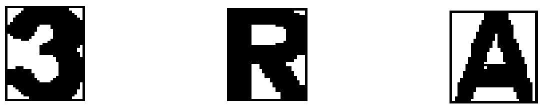
step 8: The index of the best match is stored as the recognized character. After recognizing the first character the next character is taken and thus after recognizing the first line, the next line is taken, and procedure from step 3 is repeated until the last line detected is empty.

4.2.3 Template shape

The template shape plays a vital role in character recognition. It decides the success or failure of the character identification. Since each font has a different shape and orientation, it is very difficult to design a universal template that can be attributed to all the characters. Some of the font shapes are shown below:



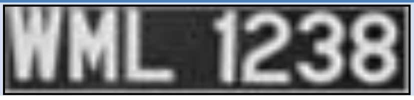
Here templates are designed in such a way that they can represent their corresponding characters. They are designed such that they possess all the characteristics related to their corresponding characters. The character templates used in this project are as shown below:



Each template is of size 42 x 24. Each of these templates is stored in the database and they are retrieved for comparison of templates during character recognition process.

If the license plates have font shapes different from the template shapes that are designed, then the character recognition may not be very effective. Such problems occur when italic fonts are used in the license plates. The algorithm will work very effectively if there is a standardization of characters.

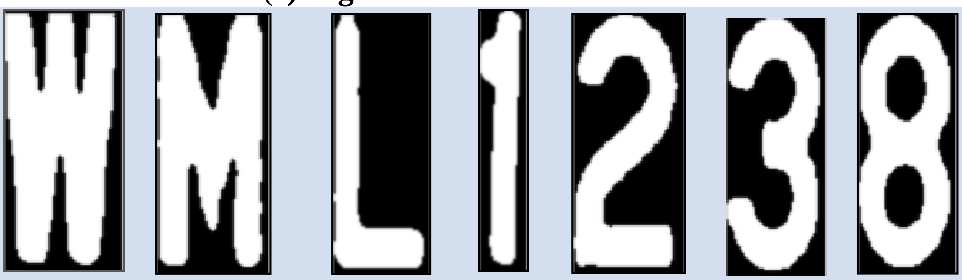
Character segmentation and extraction



(a) Gray image of extracted license plate

(b) **Image after pre processing**

(c) **Segmentation of Characters**



(d)Extracted Character

Figure 4. 4: Results of template matching

Figure 4.4 shows the result of template matching based OCR. Here (a) is extracted license plate which is the output of previous section, first pre-processing is done to enhance the image contrast, sharpening the edges and converting it into the binary image, (b) shows the result after pre-processing. (c) Segmentation of character (d) Extracted character. Some more results are shown in Figure 4.5

Figure 4. 5: some more results of character extraction

4.2.4 Measure of Effectiveness

$$Read\ rate = \frac{number\ of\ license\ plates\ accurately\ read}{total\ number\ of\ license\ plates\ tested}$$

(4.2)

Read rate represents accuracy of reading and processing characters.

Table4. 1 Read rate for character recognition

Type of license plates	Number of images	Number of license plates accurately read	Read rate in %
Standard license plates	20	18	90
License plates with good contrast	15	13	86.66
License plates with Proper lighting condition	20	18	86.66
License plates with Low resolution	15	12	80
Skewed License plates	10	6	60
Non-standard size of license plates	10	0	0

Over all read rate for the algorithm is 72.2%. Template matching algorithm gives better results if the source is cooperative.

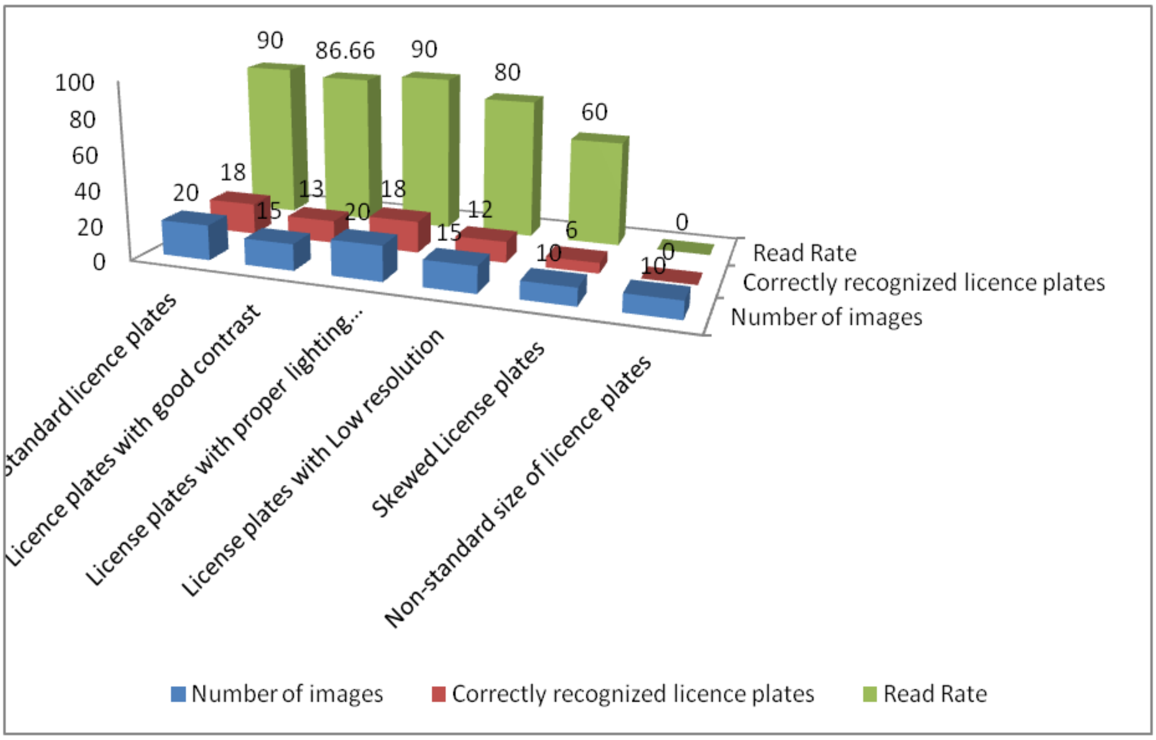


Figure4. 1: Measure of effectiveness for Template Matching

For better performance of algorithm Euler number is calculated for each character and if we consider the format of the license plate then accuracy is prominently increases as well the run time of the algorithm is also reduces.

4.2.5 Template Matching using Euler Number

In order to improve the accuracy for the above technique, topological properties such as Euler number is used. It is defined as the total number of connected component in an image, minus the number of holes in them.

$$E = C - H \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textbf{(4.3)}$$

Where C is the number of connected components and H is the number of object holes in an image.

By calculating the Euler number of an image, it is possible to distinguish between three main sets of characters:

i)Characters without holes:

Euler number = 1

ii) Characters with one hole:

Euler number = 0

iii) Characters with two holes:

Euler number = -1

While determining the character with maximum correlation in Template matching using inner-product technique, segmented character's Euler number is also checked with that of the character templates stored, to filter out those characters whose Euler numbers are not matching.

4.3 Considering the format of number plate

In order to maximize the accuracy of the system format of the Registration number is considered. Our system is design to read the license plates of India, so considering the format of Indian license plate.

4.3.1. Format of Registration Number in India

Because the project deals with the Recognition of number plates in India, the general format for all the vehicles in this country only is considered. The new format of the Registration, is as shown below:

Case (i): SS DD AA NNNN

Where *SS* is the two letter state code; *DD* is the two digit district code; *NNNN* is the unique license plate number and *AA* are the optional alphabets if the 9999 numbers are used up.

The National Capital Territory of Delhi has an additional code in the registration code:

Case (ii): SS DD C AA NNNN

Where *SS* is the two letter code for Delhi (DL) and the additional *C* is for the category of vehicle.

4.3.2. Classification of characters in License Plate

Rather than comparing each of the characters segmented, from the license plate, with all the 36-characters (26 alphabets + 10 digits) in the database; an efficient way of comparing templates is to use the prior knowledge of the Registration Number format to significantly reduce the number of computations. Therefore the templates are classified in accordance to the registration format.

4.3.2.1 Combinations of Characters

The various combinations of characters in Registration Number are shown in the table below:

Case (i):

Table4. 2 Combination of characters

No. of Characters	Combinations of Characters	1 2	3 4	5 6	7 8 9 10
10	2A + 2N + 2A + 4N	A A	N N	A A	N N N N
9	2A + 2N + 2A + 3N 2A + 2N + 1A + 4N	A A A A	N N N N	A A A N	N N N N N N
8	2A + 2N + 2A + 2N 2A + 2N + 1A + 3N 2A + 2N + 0A + 4N	A A A A A A	N N N N N N	A A A N N N	N N N N N N
7	2A + 2N + 2A + 1N 2A + 2N + 1A + 2N 2A + 2N + 0A + 3N	A A A A A A	N N N N N N	A A A N N N	N N N
6	2A + 2N + 1A + 1N 2A + 2N + 0A + 2N	A A A A	N N N N	A N N N	

Where ‘A’ represents Alphabets and ‘N’ represents Numbers

It is observed that the first two sets of the Registration Number are Alphabets and Numbers respectively. The fifth letter is an alphabet in most of the cases. Hence it is assumed as an alphabet in the algorithm. The sixth letter can be an Alphabet or a Number. The remaining letters are Numbers.

Case (ii):

For the National Capital Territory of Delhi

11- characters: 2A + 2N + 1A + 2A + 4N

The fifth letter is an Alphabet, sixth and seventh is Alphanumeric and the remaining are Numbers.

4.3.2.2 Grouping of Alphabets and Numbers

Taking into consideration the position of Alphabets and Numbers occurring in the license plate, the computations involved in comparing the templates with all 36-characters can be reduced by ‘Grouping the License Plate into a set of Alphabets and Numbers’.

Hence the first two letters in the license plate are checked for Alphabets, the next two for Numbers, the fifth for Alphabet again, the sixth for Alpha-numeric and the remaining for Numbers.

The computations involved in comparing the templates, for Case (i), are shown in the table below:

For the format: SS- DD-AA-NNNN:

Table 4. 3 Comparison of Templates

Position of Letter	Alphabet or Numerical	Comparison of Templates
1 (S)	Alphabet	26
2 (S)	Alphabet	26
3 (N)	Numeric	10
4 (N)	Numeric	10
5 (A)	Alphabet	26
6 (A)	Alpha-Numeric	36
7 (N)	Numeric	10
8 (N)	Numeric	10
9 (N)	Numeric	10
10 (N)	Numeric	10

The Total number of comparisons involved is 174.

(26+26+10+10+26+36+10+10+10+10)

This is far lesser than the comparisons involved when 'Grouping of Alphabets and Numbers' is not considered, i.e., 360 (36*10).

4.3.2.3 Minimizing the Computations

Designing the algorithm for specific applications can further reduce the number of comparisons of templates. Instead of considering all the 26 letters in the Alphabets and all the 10 letters in the Numbers, the letter that doesn't appear in the license plate can be avoided during comparisons. The following sections deals with the method of reducing the comparisons based on the prior information of Registration Marks.

4.3.2.4 Identification of the two letter State Code (SS)

All Indian states and Union Territories have their own two lettering.

Table 4. 4: Registration Marks allotted to States and Union Territories in India are:

State	Two-letter Code	State	Two-letter Code
Andhra Pradesh	AP	Mizoram	MZ
Arunachal Pradesh	AR	Nagaland	NL
Assam	AS	Orissa	OR
Bihar	BR	Punjab	PB
Chhattisgarh	CG	Rajasthan	RJ
Delhi	DL	Sikkim	SK
Goa	GA	Tamil Nadu	TN
Gujarat	GJ	Tripura	TR
Haryana	HR	Uttarakhand	UA
Himachal Pradesh	HP	Uttar Pradesh	UP
Jammu Kashmir	JK	West Bengal	WB
Jharkhand	JH	Andaman & Nicobar	AN
Karnataka	KA	Chandigarh	CH
Kerala	KL	Dadra and Nagar Haveli	DN
Madhya Pradesh	MP		
Maharashtra	MH	Daman & Diu	DD
Manipur	MN	Delhi	DL
Meghalaya	ML	Lakshadweep	LD
		Pondicherry	PY

Table 4.4 Registration marks allotted to states and UTs

Grouping of characters

I). Grouping of Alphabets based on Registration Marks

The following alphabets appear in the first position of the License Plate:

Group (1): A, B, C, D, G, H, J, K, L, M, N, O, P, R, S, T, U and W.

Total number of letters in Group (1) is 18.

The following alphabets appear in the second position of the License Plate:

Group (2): A, B, D, H, J, K, L, N, P, R, S, Y and Z.

Total number of letters in Group (2) is 13.

Therefore only these groups are checked for the first and second positions.

The number of comparisons involved for the first two letters is: 31 (18+13).

The number of comparisons involved for the first two letters, if ‘Grouping based on Registration Marks’ is not considered, is: 52 (26+26).

The number of comparisons involved for the first two letters, if ‘Grouping into sets of Alphabets and Numbers’ is not considered, is: 72 (36+36).

II). Recognition of Second letter based on the first

The number of computations can be reduced further if the order of the appearance of the letters in the State Code is strictly considered, i.e., the probable choice of the second letter can be inferred once the first letter is identified. The probable choice of the second letter, when the first letter is known, is shown in the table 4.5 below:

Table 4. 5 Choice of second letter

First Letter	Probable choice of Second Letter	First Letter	**Probable choice of Second Letter**
A	N, P, R, S	M	H, L, N, P, Z
B	R	N	L
C	H	O	R
D	D, L, N	P	B, Y
G	A, J	R	J
H	P, R	S	K
J	K	T	N, R
K	A, L	U	P
L	D	W	B

If the first letter is identified as ‘M’ then the number of comparisons involved in the identification of the second letter are 5 instead of 13, as in Group (2).

If the first letter is identified as ‘R’ then the number of comparisons involved in the identification of the second letter is only ‘one’.

The main drawback with this method lies with the heavy reliance of the first letter in the identification of second letter. If the first letter is recognized incorrectly then there are good chances in the fallibility of the second letter.

Chapter -5

Conclusion and Future work

Here in this thesis, the methods for traffic surveillance have been presented and the work on motion detection, license plate extraction and character recognition is carried out. In motion detection, a study on different background subtraction available in the literature has been studied and their performance tests on the different video test sequence are given. The fitness coefficient and error coefficient is also calculated for all the methods. It should be noted that robust motion detection is a critical task and its performance is affected by the presence of varying illumination, background motion, camouflage, shadow, and etc.

In license plate extraction the strength and weakness of the different extraction algorithm have discussed which are available in the literature and comparisons of all the methods have been done. Proposed two methods for extraction of license plate i.e edge detection method and the block variance technique are presented. The block variance algorithm has been tested on 90 images and giving 87.4% accuracy measure.In character extraction template matching (OCR) algorithm is used for extraction and different algorithms that are presented in literature survey are also studied. For improving the performance of template matching algorithm the format of license plate is studied.

This integrated system locates tracks and extracts traffic parameters in real time. Furthermore, the system can utilize any existing traffic surveillance infrastructure without further modification or tuning (except for the camera calibration that calculates image metrics). Overall, the system was found to work satisfactorily and the background reconstruction algorithm added robustness to the process. In normal traffic conditions the system responded well and the outcome results regarding vehicle license plate and trajectory were accurate enough. The experiments carried out showed that the proposed algorithm is capable of real time operational working due to its low complexity.The background reconstruction algorithm allows the unobstructed operation of the system without human intervention. The system works well either in real time mode or in already stored video.

In future work, we aim to focus on night surveillance and to improve the existing algorithms reported in literature. However, the other segments of our suggested system should be improved, focusing on the occlusion handling, vehicle matching procedure and also focus on improving the accuracy measure for character recognition by using the concept of neural network for recognising all font type of a character by using back propagation algorithm. In this, first the network is trained and to train the network, the input and target are required. After the network had been successfully trained, the segmented character in license plate can now inputted the neural network to simulation. Ideally, the input characters will compare with the data that trained in neural network, and then outputted the ASCII code for corresponding input character.

Bibliography

[1] Hu, Weiming, Tieniu Tan, Liang Wang, and Steve Maybank. "A survey on visual surveillance of object motion and behaviors." *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 34, no. 3 (2004): 334-352.

[2] Buch, Norbert, Sergio A. Velastin, and James Orwell. "A review of computer vision techniques for the analysis of urban traffic." *Intelligent Transportation Systems, IEEE Transactions on* 12, no. 3 (2011): 920-939.

[3] Zhu, Zhongjie, and Yuer Wang. "A hybrid algorithm for automatic segmentation of slowly moving objects." *AEU-International Journal of Electronics and Communications* 66, no. 3 (2012): 249-254.

[4] Lipton, Alan J., Hironobu Fujiyoshi, and Raju S. Patil. "Moving target classification and tracking from real-time video." In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pp. 8-14. IEEE, 1998.

[5] Barron, John L., David J. Fleet, and S. S. Beauchemin. "Performance of optical flow techniques." *International journal of computer vision* 12, no. 1 (1994): 43-77.

[6] Barnich, Olivier, and Marc Van Droogenbroeck. "ViBe: A universal background subtraction algorithm for video sequences." *Image Processing, IEEE Transactions on* 20, no. 6 (2011): 1709-1724.

[7] Elgammal, Ahmed, David Harwood, and Larry Davis. "Non-parametric model for background subtraction." In *Computer Vision—ECCV 2000*, pp. 751-767. Springer Berlin Heidelberg, 2000.

[8] Toyama, Kentaro, John Krumm, Barry Brumitt, and Brian Meyers. "Wallflower: Principles and practice of background maintenance." In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 255-261. IEEE, 1999.

[9] Panda, Deepak Kumar. "Motion detection, object classification and tracking for visual surveillance application." PhD diss., 2012.

[10] Haritaoglu, Ismail, David Harwood, and Larry S. Davis. "W4: real-time surveillance of people and their activities." Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, no. 8 (2000): 809-830.

[11] Antoine Manzanera and Julien C. Richefeu. A new motion detection algorithm based on [Sigma]-[Delta] background estimation. Pattern Recognition Letters, 28(3):320–328, February 2007.

[12] Manuel Vargas. et al."An Enhanced Background Estimation Algorithm for Vehicle Detection in Urban Traffic Scenes", IEEE Transactions On Vehicular Technology, Vol. 59, No. 8, October2010.

[13] Stauffer, C., & Grimson, W. (1999). Adaptive background mixture models for real-time tracking.Computer Vision Pattern Recognition, 246–252.

[14] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in Proc. Eur. Conf. Comput. Vis., May 2002, vol. 3, pp. 543–560.

[15] Wei Zhang. et al. “Moving vehicles segmentation based on Bayesian framework for Gaussian motion model” Pattern Recognition Letters 27 (2006) 956–967, ELSEVIER

[16] Cucchiara, R., Piccardi, M. (1999). Vehicle detection under day and night illumination. In Proceedings of 3rd international ICSC symposium on intelligent industrial automation.

[17] Z. Zhu and G. Xu, “VISATRAM: A real-time vision system for automatic traffic monitoring,” Image Vis. Comput., vol. 18, no. 10, pp. 781–794, Jul. 2000.

[18] D.Panda and S.Mehar “Robust Real-Time Object Tracking Under Background Clutter”, International Conference on Image Information Processing (ICIIP 2011) 2011.

[19] Nicholas A., Iphigenia K.& Chris T. Kiranoudis. A background subtraction algorithm for detecting and tracking vehicles. ELSEVIER Expert Systems with Applications 38 (2011) 1619–1631

[20] K.K. Kim. et al. "Learning-based approach for license plate recognition," IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing, vol. 2, pp. 614- 623.2000.

[21] Aditya Acharya and Sukadev Meher “Region Adaptive Unsharp Masking based DCT Interpolation for Efficient Video Intra Frame Up-Sampling.” IJCA Special Issue on International Conference ICEDSP(3):24-28, February 2013.

[22] Hongwei Ying, Jiatao Song, Xiaobo Ren” Character Segmentation for License Plate by the Separator Symbol's Frame of Reference” 2010 International Conference on Information, Networking and Automation (ICINA).

[23] “Mathematical morphology” from Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Mathematical_morphology#Closing

[24] Shuang Qiaol , Yan Zhul , Xiufen Li l , Taihui Liu2 ,3, Baoxue Zhangl “Research of improving the accuracy of license plate character segmentation” 2010 Fifth International Conference on Frontier of Computer Science and Technology.

[25] Deng Hongyao, Song Xiuli “License Plate Characters Segmentation Using Projection and Template Matching”, 2009 International Conference on Information Technology and Computer Science.

[26] Yungang Zhang, Changshui Zhang, “A New Algorithm for Character Segmentation of License Plate”: Dept. of Automation, Tsinghua University, The Institute of Information Processing Beijing, Chin

[27] Huang, Yo-Ping, Chien-Hung Chen, Yueh-Tsun Chang, and Frode Eika Sandnes. "An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition." Expert Systems with Applications 36, no. 5 (2009): 9260-9267.

[28] Shapiro, Vladimir, Georgi Gluhchev, and Dimo Dimov. "Towards a multinational car license plate recognition system." *Machine Vision and Applications* 17, no. 3 (2006): 173-183.

[29] Efford , Nick. "Digital Image Processing: A Practical Introduction Using Java. 2000."

[30] Gonzalez, Rafael C., Richard E. Woods, and Steven L. Eddins. *Digital image processing using MATLAB*. Vol. 2. Tennessee.,Gatesmark Publishing, 2011